



## โครงการ

# การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ แฮชแท็กเกอร์

Hashtagor

ชื่อนิสิต	นางสาวขวัญชนก	ศรีสมพงษ์	583 36127 23
	นายชนินทร์ชัย	หาญเมือง	583 36162 23

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์  
สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2561

**คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย**

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของโครงการงานทางวิชาการที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของโครงการงานทางวิชาการที่ส่งผ่านทางคณะที่สังกัด

The abstract and full text of senior projects in Chulalongkorn University Intellectual Repository(CUIR)  
are the senior project authors' files submitted through the faculty.

แฮชแท็กเกอร์

นางสาวขวัญชนก ศรีสมพงษ์  
นายชินทร์ชัย หาญเมือง

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต  
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย  
ปีการศึกษา 2561  
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

# HASHTAGOR

Kwanchanok Srisompong  
Chaninchai Hanmuang

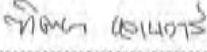
A Project Submitted in Partial Fulfillment of the Requirements  
for the Degree of Bachelor of Science Program in Computer Science  
Department of Mathematics and Computer Science  
Faculty of Science  
Chulalongkorn University  
Academic Year 2018  
Copyright of Chulalongkorn University

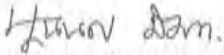
หัวข้อโครงการ	แอสแท็กเกอร์
โดย	นางสาวขวัญชนก ศรีสมพงษ์ นายชนินทร์ชัย หาญเมือง
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงการหลัก	ผู้ช่วยศาสตราจารย์ ดร. จิตยา หวานนารี

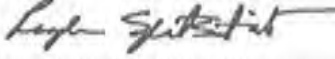
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย  
อนุมัติให้นับโครงการฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา  
2301499 โครงการวิทยาศาสตร์ (Senior Project)

  
.....  
(ศาสตราจารย์ ดร. กฤษณะ เนียมมณี) หัวหน้าภาควิชาคณิตศาสตร์  
และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงการ

  
.....  
(ผู้ช่วยศาสตราจารย์ ดร. จิตยา หวานนารี) อาจารย์ที่ปรึกษาโครงการหลัก

  
.....  
(ศาสตราจารย์ ดร. ยศนันต์ มีมาก) กรรมการ

  
.....  
(รองศาสตราจารย์ ดร. พิชะพนธ์ โสพิศสถิตย์) กรรมการ

ขวัญชนก ศรีสมพงษ์, ชนินทร์ชัย หาญเมือง: แฮชแท็กเกอร์. (Hashtagor) อ.ที่ปรึกษา  
 โครงการหลัก: ผศ. ดร. จิตยา หวานวารี, 79 หน้า.

โครงการ เรื่อง “แฮชแท็กเกอร์” เป็นโครงการที่จัดทำขึ้นเพื่อช่วยสังเคราะห์แฮชแท็ก  
 ในข้อความให้กับผู้ใช้บนทวิตเตอร์ประเทศไทย เนื่องจากผู้พัฒนาเห็นปัญหาว่า การคิดแฮชแท็ก  
 ในบางครั้งนั้นไม่ง่าย เนื่องจากคำที่ใช้ในแฮชแท็กอาจไม่เหมาะสมกับเนื้อหาหรืออาจไม่เชื่อมโยงถึง  
 ประเด็นที่กำลังเป็นที่นิยมบนทวิตเตอร์ประเทศไทย ผู้พัฒนาจึงพัฒนาโปรแกรมประยุกต์  
 บนโทรศัพท์เคลื่อนที่สำหรับระบบปฏิบัติการไอโอเอส

ในส่วนของตัวโปรแกรมประยุกต์มี 2 ฟังก์ชันหลักคือ ฟังก์ชันสังเคราะห์แฮชแท็กใหม่จาก  
 ข้อความทวิตของผู้ใช้และฟังก์ชันแนะนำแฮชแท็กที่เป็นที่นิยมในประเทศไทยที่สอดคล้องกับข้อความ  
 ทวิตมากที่สุด

จากการพัฒนาโปรแกรมประยุกต์ “แฮชแท็กเกอร์” ผู้พัฒนาได้ทำการทดสอบโปรแกรม  
 ทั้งสองฟังก์ชันหลักสำหรับฟังก์ชันสังเคราะห์แฮชแท็กใหม่ได้ทดสอบกับนิสิตคณะวิทยาศาสตร์  
 จุฬาลงกรณ์มหาวิทยาลัย จำนวน 100 คนพบว่า ร้อยละ 70 จากทั้งหมดพอใจกับผลลัพธ์ของฟังก์ชัน  
 นี้และผลการทดสอบฟังก์ชันแนะนำแฮชแท็กที่เป็นที่นิยมในประเทศไทย เมื่อทดสอบกับชุดข้อมูล  
 ทดสอบพบว่ามียอดการค้นคืนถึงร้อยละ 78 ซึ่งนับว่าเป็นที่น่าพอใจ

ภาควิชา...คณิตศาสตร์และวิทยาการคอมพิวเตอร์...ลายมือชื่อนิสิต... *ขวัญชนก ศรีสมพงษ์*  
 ลายมือชื่อนิสิต... *ชนินทร์ชัย หาญเมือง*  
 สาขาวิชา...วิทยาการคอมพิวเตอร์...ลายมือชื่อ อ.ที่ปรึกษาโครงการหลัก... *จิตยา หวานวารี*  
 ปีการศึกษา... 2561...ลายมือชื่อ อ.ที่ปรึกษาโครงการร่วม...

# # 5833612723, 5833616223: MAJOR COMPUTER SCIENCE

KEYWORDS: HASHTAG/TRENDING/GENERATING


KWANCHANOK SRISOMPONG, CHANINCHAI HANMUANG:

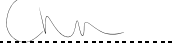
HASHTAGOR. ADVISOR: ASST. PROF. DITTAYA WANVARIE, Ph.D., 79 pp.

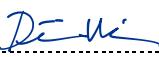
This "Hashtagor" project is a mobile application which can synthesize hashtags from a tweet for users on Twitter. We have found that it was sometimes difficult for a user to create an appropriate hashtag for the target tweet since words in the hashtag might not be related to the tweet. Moreover, the hashtag may not be a currently popular hashtag on Twitter in Thailand. As a result, we have developed a mobile application running on the iOS operating system to solve the problem.

There are two main functions in the application; the first function is to generate hashtags from a user's tweet. The other function is to recommend trending hashtags, which are mostly related to the user's tweet.

We have evaluated our application in two folds. For the generating hashtags function, we have asked 100 students from the Faculty of Science, Chulalongkorn University, for the satisfaction of the results. The results show that 70% of the testers are satisfied with the generated hashtags results. The other evaluation is for the trending hashtags model; the results show that we achieve validation recall rate at 76.51%.

Department: Mathematics and Computer Science Student's Signature 

Student's Signature 

Field of Study: Computer Science Advisor's Signature 

Academic Year: 2018 Co-advisors Signature \_\_\_\_\_

## **ACKNOWLEDGMENTS**

Developers would like to express our very great appreciation to Asst. Prof. Dr. Dittaya Wanvarie and Dr. Naruemon Pratanwanich for their valuable and constructive suggestions during the planning and development of this project. Their willingness to give their time so generously has been very much appreciated.

We would also like to thank The National Electronics and Computer Technology Centre National Science and Technology Development Agency and Ministry of Science and the Department of Mathematics and Computer Science Faculty of Science, Chulalongkorn University, for giving us grants for this project also a chance to participate in the Twenty-First National Software Contest, NECTEC, Thailand: NSC21 in Mobile Application category.

# CONTENTS

	Page
<i>ABSTRACT IN THAI</i> .....	<i>iv</i>
<i>ABSTRACT IN ENGLISH</i> .....	<i>v</i>
<i>ACKNOWLEDGMENTS</i> .....	<i>vi</i>
<i>CONTENTS</i> .....	<i>vii</i>
<i>LIST OF TABLES</i> .....	<i>viii</i>
<i>LIST OF FIGURES</i> .....	<i>ix</i>
<b>CHAPTER I INTRODUCTION</b> .....	<b>1</b>
1.1 Background and rationale .....	1
1.2 Objectives .....	2
1.3 Scope.....	2
1.4 Project Activities .....	3
1.5 Benefits .....	3
1.6 Report Outlines .....	4
<b>CHAPTER II RELATED WORKS</b> .....	<b>5</b>
2.1 Word Embedding - Skip-Gram Model.....	5
2.2 Long-Short Term Memory .....	5
2.3 Quora Question Pairs using Siamese Manhattan LSTM .....	6
2.4 Generating News Headline with Recurrent Neural Networks.....	7
2.5 React Native .....	10
2.6 Expo SDK.....	11
2.7 Client/Server Architecture .....	11
<b>CHAPTER III APPLICATION</b> .....	<b>12</b>
3.1 Storyboard and Software Design .....	12
3.2 Dataflow diagram.....	22
3.3 Usecase diagram .....	23
3.4 Software Architecture .....	25
3.5 Software Specification.....	25
<b>CHAPTER IV RESULTS</b> .....	<b>32</b>
4.1 Model Testing Results .....	32
4.2 Application Testing Results.....	37
<b>CHAPTER V CONCLUSION</b> .....	<b>40</b>
5.1 Conclusion .....	40
5.2 Suggestion .....	40
<b>REFERENCES</b> .....	<b>42</b>
<b>APPENDIX A The Project Proposal of Course 2301399 Project Proposal Academic Year 2061</b> .....	<b>45</b>
<b>APPENDIX B USER'S MANUAL</b> .....	<b>60</b>



## LIST OF TABLES

	Page
Table 1.1 Project Activity.....	3
Table 3.1 Usecase: Submit an input text .....	24
Table 3.2 Usecase: View list of hashtags .....	24
Table 3.3 Usecase: Select or deselect hashtags .....	24
Table 3.4 Usecase: Copy the input text with selected hashtags .....	24

## LIST OF FIGURES

	Page
Figure 2.1 LSTM .....	6
Figure 2.2 Siamese Manhattan LSTM Model .....	7
Figure 2.3 Encoder-decoder neural network architecture .....	7
Figure 2.4 Simple attention .....	9
Figure 3.1 Splash page .....	12
Figure 3.2 Operation page .....	13
Figure 3.3 Please input text message dialog.....	14
Figure 3.4 Miss word message dialog .....	15
Figure 3.5 Text too long message dialog example.....	16
Figure 3.6 Inserted input example.....	17
Figure 3.7 Results page example.....	18
Figure 3.8 Selected hashtags example.....	19
Figure 3.9 Copy to clipboard example .....	20
Figure 3.10 Continue using example.....	21
Figure 3.11 Hashtagor's dataflow diagram.....	22
Figure 3.12 Hashtagor's usecase diagram.....	23
Figure 3.13 Software Architecture .....	25
Figure 3.14 Sequence to sequence model.....	27
Figure 3.15 Visualization of Sequence to Sequence Model for generating hashtags ..	28
Figure 3.16 Trending Hashtags Model.....	30
Figure 3.17 Visualization of Trending Hashtags Model .....	30
Figure 4.1 Dataset example .....	32
Figure 4.2 Hashtags and scores1 .....	33
Figure 4.3 Hashtags and scores2 .....	34
Figure 4.4 Data example.....	35
Figure 4.5 Model recall .....	35
Figure 4.6 Model loss .....	36
Figure 4.7 Before hit operation button example.....	37
Figure 4.8 Result after hit Operation button example .....	38
Figure 4.9 Result after hit Copy button example .....	39
Figure B.1 User's manual: Splash page .....	60
Figure B.2 User's manual: Operation page component .....	61
Figure B.3 User's manual: Inserted input example .....	62
Figure B.4 User's manual: Please input text message dialog.....	63
Figure B.5 User's manual: Miss word message dialog .....	64
Figure B.6 User's manual: Text too long message dialog example.....	65
Figure B.7 User's manual: Results page example.....	66
Figure B.8 User's manual: Selected hashtags example.....	67
Figure B.9 User's manual: Copy to clipboard success example .....	68
Figure B.10 User's manual: Continue using example.....	69

# CHAPTER I

## INTRODUCTION

### 1.1 Background and rationale

For humans, social interaction is an indispensable issue. Because human is a social animal that needs to communicate with each other. Communication tools have evolved from a dove, a letter, fax, to the era of social networking that plays a vital role in communication nowadays. Especially in Thailand, the country that has the highest average number of daily hours spent on the internet among countries in the world [1]. The most current applications are Facebook, Twitter, and Instagram which are also used by a variety of age groups. At present, communication among people becomes more complicated. Apart from the information from a conventional media source, an ordinary user can also provide his or her information to the public. These overwhelming data make it challenging to find a suitable piece of information for the target social groups. Therefore, a hashtag or # symbol is created to help people with a common interest in information on the same topic. Another application of a hashtag is helping people to find the trending topic at that time. For example, hashtags are heavily used in a message (or Tweet) from Twitter. A tweet, “ลองไหว้ชั้นสวยๆสักทีสิ ชั้นอาจจะให้อีกสัก 2 3 ล้าน #แรงเงา” has a hashtag “แรงเงา” which users can click on the hashtag to see other tweets with the same hashtag from other people.

Sometimes, finding an appropriate hashtag is not easy. Since users may use any words in a hashtag, the selected words may not be suitable for the tweet content. Moreover, the user created hashtag may not be popular, which make it difficult for other people to discuss the same topic [2]. Different users may come up with different hashtags although their tweets are on the same topic. However, if users know the popular hashtags or hot issues on Twitter at that time, they can create a new hashtag that is related to their tweets. The new hashtag may become a new trending hashtag. From the stated problems, our application provides a user with hashtags based on the tweet content from the following two processes:

1. The synthesis of hashtags based on messages in the user's tweet.
2. The selection of related hashtags that are currently in the top 100 trending topics in Thailand tweet.

## **1.2 Objectives**

1. The application must be able to generate new hashtags and recommend trending hashtags in the Thailand region.
2. The application must be easy to use, support iOS and be able to copy user's selected results to the clipboard.

## **1.3 Scope**

1. Only supports the iOS operating system.
2. Can copy input messages with selected hashtags to the clipboard.
3. Use messages in Tweet that are trending in Twitter Thailand.
4. Use only Thai or English text.

## 1.4 Project Activities

**Table 1.1 Project Activity**

Activity	2018					2019			
	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
1. Study theory on Deepcut[3], Tweepy[4], Pandas[5], TensorFlow[6], and Word2Vec[7]									
2. Collect data									
3. Analyze data and develop the model									
4. Study on creating an application									
5. Design and develop the application									
6. Test, discuss and conclude the results									
7. Prepare documentation									

## 1.5 Benefits

### 1.5.1 Benefits of the project from a user aspect

1. Save time in the selection or creation of a hashtag.
2. Can be used for marketing purpose.
3. Enjoy the program.
4. Make the message more popular.
5. Make users know topics that are currently popular on Twitter.

### **1.5.2 Benefit for students who implement this project.**

1. Gain knowledge and understanding of work procedures. Practice thinking, analyzing, working in a structured manner. Know how to work as a team with discipline, punctuality, and responsibility for work.
2. Gain experience in developing a mobile application on iOS operating systems, including the knowledge on natural language processing.
3. Understand how to extract data from Twitter using the API.
4. Develop skills in Python programming and learn how to use different libraries.

## **1.6 Report Outlines**

Chapter 2 introduces the theories, technologies, techniques, algorithms, and tools that are used in this project.

Chapter 3 shows storyboard; software design and software specification of the application including tools and library that are used in this project.

Chapter 4 shows the details of the datasets used, model testing results and application testing results.

Chapter 5 concludes the project.

## **CHAPTER II**

### **RELATED WORKS**

This chapter introduces the readers to the theories, technologies, techniques, algorithms, and tools used in this project.

#### **2.1 Word Embedding – Skip - Gram Model**

There are 2 purposes of skip-gram [8]. The first one is to characterize a selected word, phrase, or sentence based on other words, phrases, or sentences around it. The other purpose is to represent each word in a corpus as a vector which aims to calculate the probability of the word showing up around another word. So, the similarities of each pair of words can be calculated by its vector.

#### **2.2 Long-Short Term Memory**

Long Short-Term Memory (LSTM) networks (see Figure 2.1) [9] is an improved version of recurrent neural networks. LSTM is a variant of RNNs, but the structure inside each node of hidden layers is different from a vanilla RNN. LSTMs implement each node of RNNs to remember inputs over a long period. It is similar to the memory because it can write, read and delete information. There is a gated cell to determine whether or not the network will store information based on its weights, which mean the importance of information. The gates in an LSTM range from 0 to 1 because of a sigmoid function.

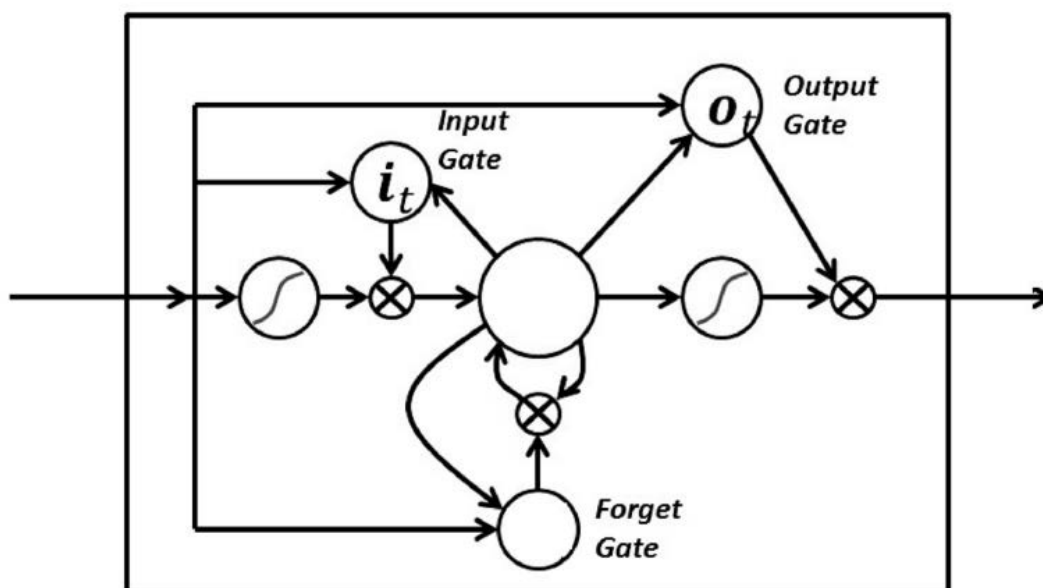


Figure 2.1 LSTM

From: <https://machinelearning-blog.com/2018/02/21/recurrent-neural-networks>

### 2.3 Quora Question Pairs using Siamese Manhattan LSTM

Siamese networks [10] are networks that have sub-networks. They perform well on similarity tasks and have been used for tasks such as sentence similarity, face recognition, etc. So, Siamese Manhattan Long Short-Term Memory is a Siamese network using LSTM and Manhattan distance (see Figure 2.2).

Quora Question Pairs is an active Kaggle Competition, which challenges participants to classify whether question pairs are duplicated or not. Elior Cohen [11] implemented the model called Siamese recurrent architectures for learning sentence similarity [10] for the competition and achieved an 82.5% accuracy rate on the validation data. Moreover, the 1st rank of the competition adopted this model to implement their model.



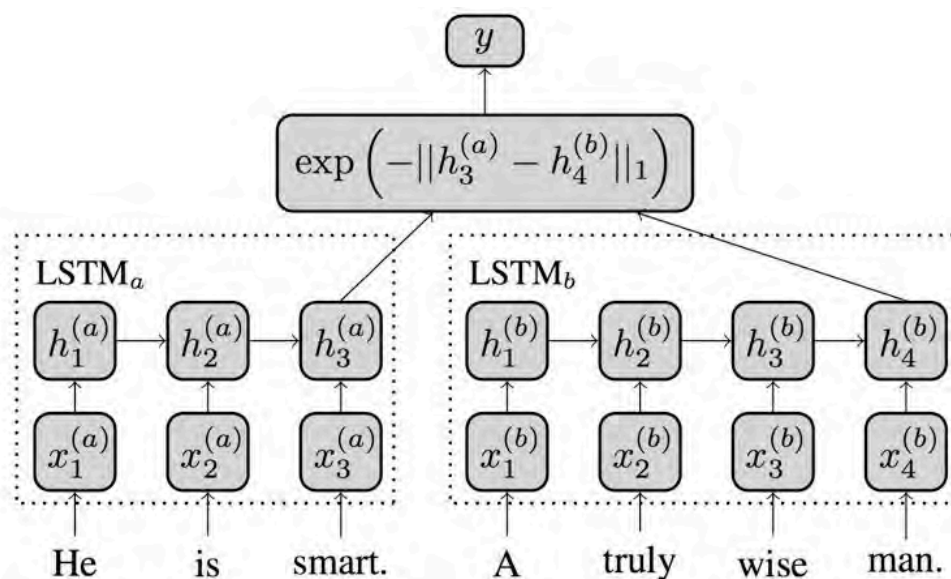


Figure 2.2 Siamese Manhattan LSTM Model

From: <https://medium.com/mlreview/implementing-malstm-on-kaggles-quora-question-pairs-competition-8b31b0b16a07>

## 2.4 Generating News Headline with Recurrent Neural Networks

Konstantin Lopyrev et al. [12] describe an encoder-decoder Recurrent Neural Networks (RNNs) with Long Short-Term Memory networks (LSTMs) and attention to generate headlines from the text of news articles. Notably,  $\langle \text{eos} \rangle$  is an end-of-sequence symbol.

### 2.4.1 Model

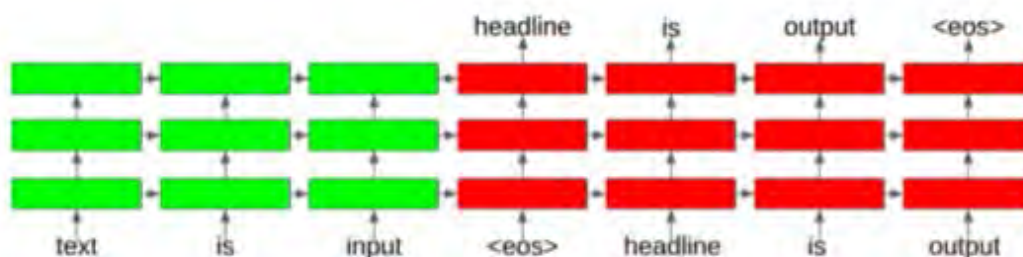


Figure 2.3 Encoder-decoder neural network architecture

From: Generating News Headlines with Recurrent Neural Networks, pg 1.

They used the encoder-decoder architecture (see Figure 2.3). The architecture includes two parts, an encoder, and a decoder. The inputs as texts are fed into the encoder part. Each word of the text is first passed through an embedding layer transforming the word into a representation, and it is then passed through a multi-layer neural network.

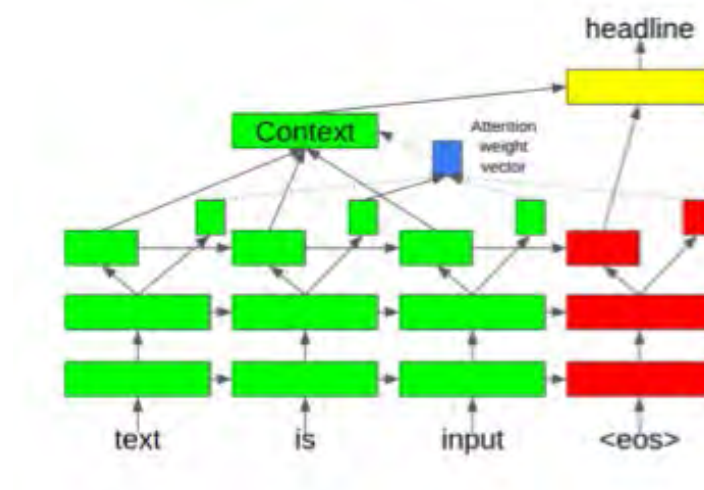
The decoder takes input from an output of a hidden unit which comes from the last word in the text of the encoder part. The input of the current unit at the decoder part accepts each output from the previously hidden unit, and the decoder then computes the probability through a softmax layer, and the attention described in the next subsection.

Finally, they used a beam-search decoder algorithm to generate words of a headline one at a time, at each step showing the scores meaning the probability from each of beam samples.

### **2.4.2 Attention**

Attention is a technique which helps Recurrent Neural Network recognize some characteristics of the input. This attention is usually used when each unit is getting output in the decoder. The attention calculates a weight for each output word over each input word to decide how much attention or importance should be spent to that input word. Then, the weights are used to calculate a weighted average referring to the context which is an input for a softmax layer.

They experiment with an attention mechanism called simple attention (see Figure 2.4). We also choose this mechanism to implement in our decoder part. They divide the hidden units of the last Long Short-Term Memory layer into two sets: one for calculating the attention weight using 50 units, and another for calculating the context using 550 units.



**Figure 2.4 Simple attention**

From: Generating News Headlines with Recurrent Neural Networks, pg 3.

### 2.4.3 Dataset

They train the model with English Gigaword dataset which is retrieved from the Stanford Linguistics department. This dataset includes news articles from 6 major news agencies in several years consisting of a headline and paragraphs. There are 5.5M news articles words with 236M words in the training data.

## **2.5 React Native**

### **2.5.1 React Native**

React Native is a mobile application framework developed by Facebook [13]. React Native can build mobile applications using only JavaScript. It uses the same design as React [14]. React Native uses the same fundamental user interface (UI) building blocks as regular iOS and Android apps, so the applications built by React Native are not mobile web applications, but a native app. React makes it easy to create interactive UIs using declarative components. A developer can design a simple view for each state and React will efficiently update and render components whenever data changes. Codes will be more predictable and more comfortable to debug with declarative views.

Furthermore, React Native lets us build an application faster than building from scratch. Instead of recompiling, we can reload an application immediately. We can even run new codes while retaining the application state. Finally, it is simple to mix React Native codes with native codes if we need to optimize some aspects of the application.

### **2.5.2 State, Props and Lifecycle**

State and Props [15] are two types of data that control a component. The state is internal to a component, while props are passed to a component. So the state is used within components to keep tracking of information. Props are immutable so components receive props from their parent and props should not be modified inside the component. All class-based components will re-render [16] themselves whenever they receive props or their state or context changes.

### **2.5.3 React Navigation**

React Navigation [17] is born from the React Native community's requirement for an extensible, yet easy-to-use navigation resolution written entirely in JavaScript. React Navigation is extensible at every layer. We can write our navigators or even replace the user-facing API. Furthermore, platform details look-and-feel with smooth animations and gestures that are completely customizable.

## **2.6 Expo SDK**

Expo is a set of tools, libraries, and services we use to build native iOS apps in a short time. The Expo SDK [18] is a set of libraries written natively for each program which provides access to the device's system functionality from JavaScript. The SDK is designed to smooth out differences in platforms as much as possible, which makes the project very transferrable because it can run in any native environment containing the Expo SDK. Expo also affords UI components to handle a diversity of use-cases that almost all apps will cover but are not built into React Native core, e.g., icons, blur views, and more.

## **2.7 Client/Server Architecture**

Client/Server Architecture [19] is a model which shows how data and processing are distributed across a variety of components that can be implemented on a single computer. A client is any process that requests specific services from server processes. A server is a process that provides requested services for clients. Both clients and servers can reside in the same computer or different computers connected by a network. Data in an interactive database has to be accessible from various locations. Because servers can be copied, it may also be used when the load on a system is mutable. The maximum gain of this model is that servers can be distributed across a network.

## CHAPTER III

### APPLICATION

This chapter shows storyboard, software design and software specification of the application including tools and library that are used in this project.

#### 3.1 Storyboard and Software Design

##### 3.1.1 Splash page

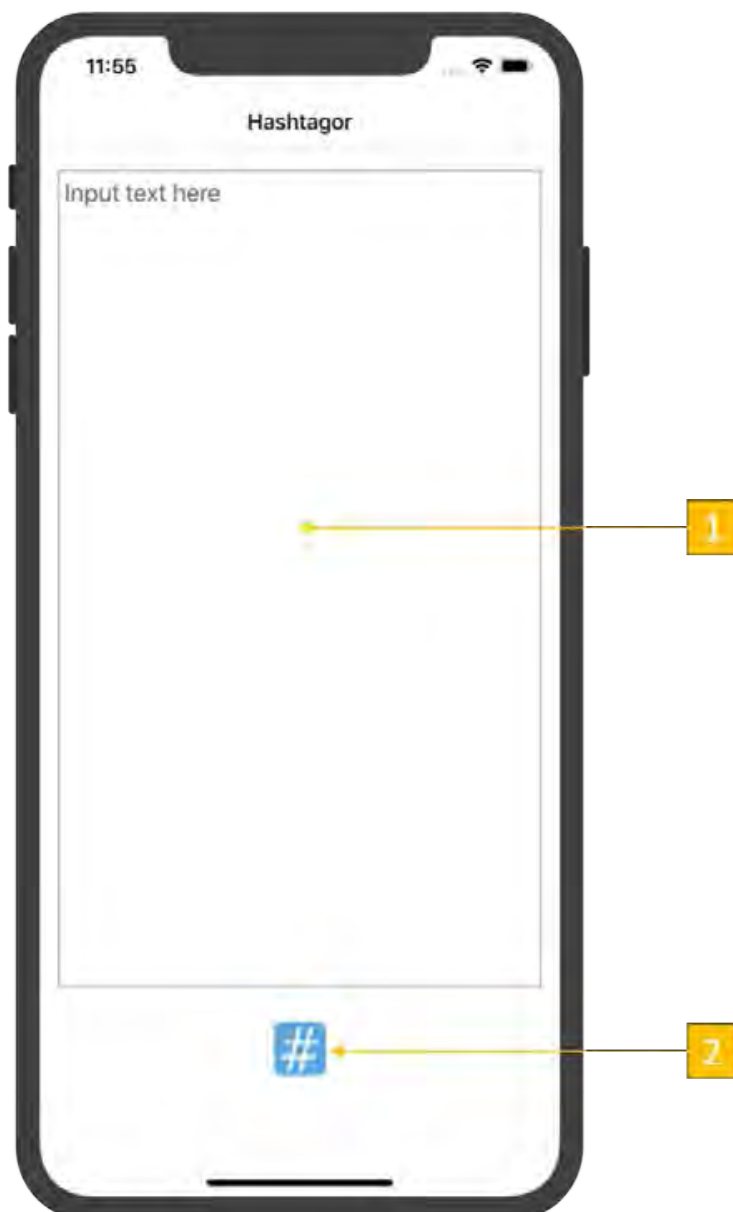
Splash page (see Figure 3.1) displays the application name while loading the application. We use a splash page to let the user know that our application is loading in case it takes time loading application to calm down the user.



Figure 3.1 Splash page

### 3.1.2 Operation page

Operation page (see Figure 3.2) is an input page that consists of a text box (see number 1 in Figure 3.2) and an operation button (see number 2 in Figure 3.2). We use almost the entire screen for a text box with text "Input text here" inside text box to let the user know that they can input text here and can see the entire input so the user can check or edit text easily. Moreover, there's only one button under the text box "operation button" whose icon is an application logo that has a Hashtag symbol inside means hitting this button to get hashtags.



**Figure 3.2 Operation page**

If the user does not input any text on the text box but hit operation button, the application will show message dialog "Please input text" (see Figure 3.3).



**Figure 3.3 Please input text message dialog**



Alternatively, if the input text does not have a word in corpus, the application will show message dialog "Miss word ..... in corpus" to notice (see Figure 3.4).



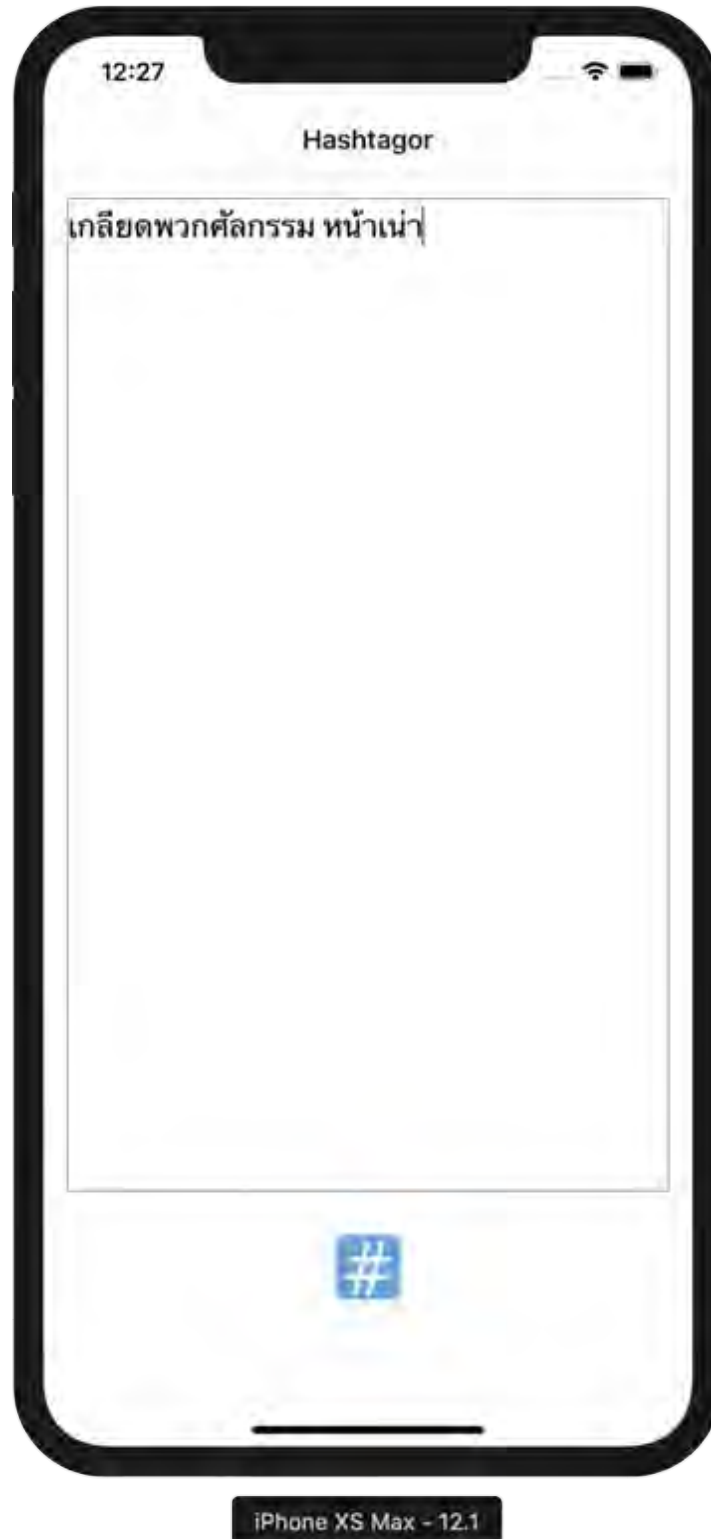
**Figure 3.4 Miss word message dialog**

Alternatively, if the input text is too long, the application will show message dialog "Text is too long" to notify the user (see Figure 3.5).



Figure 3.5 Text too long message dialog example

After entering text (see Figure 3.6) and pressing the operation button the application will be processing the Hashtags that correspond to the message.



**Figure 3.6** Inserted input example

### 3.1.3 Results page

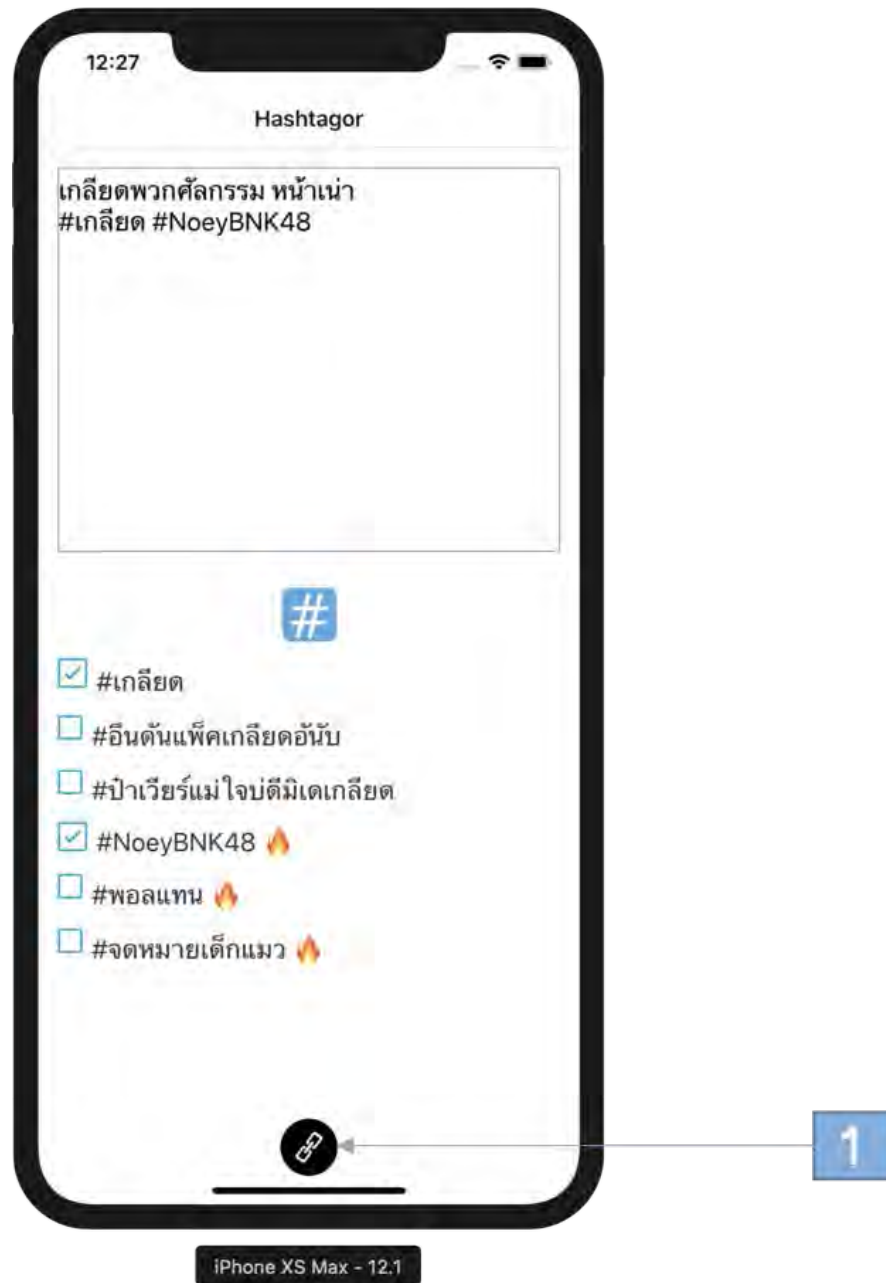
After pressing the operation button, the application will return 6 Hashtags (see Figure 3.7) shown below the operation button as a checkbox list.



**Figure 3.7 Results page example**

The first three hashtags are the Hashtags that are processed from the text only, and the latter three hashtags with fire icon are the popular Hashtags at that time and are consistent with the input messages.

User can select and deselect multiple desired Hashtags. The selected Hashtags have a check icon and are shown on the textbox following input text which is similar to a standard format on twitter (see Figure 3.8). User can copy the text and the selected Hashtag to the clipboard by clicking the Copy button (see number 1 in Figure 3.8).



**Figure 3.8 Selected hashtags example**

The application will show message dialog “Copied to clipboard!” when the copying process finished (see Figure 3.9).

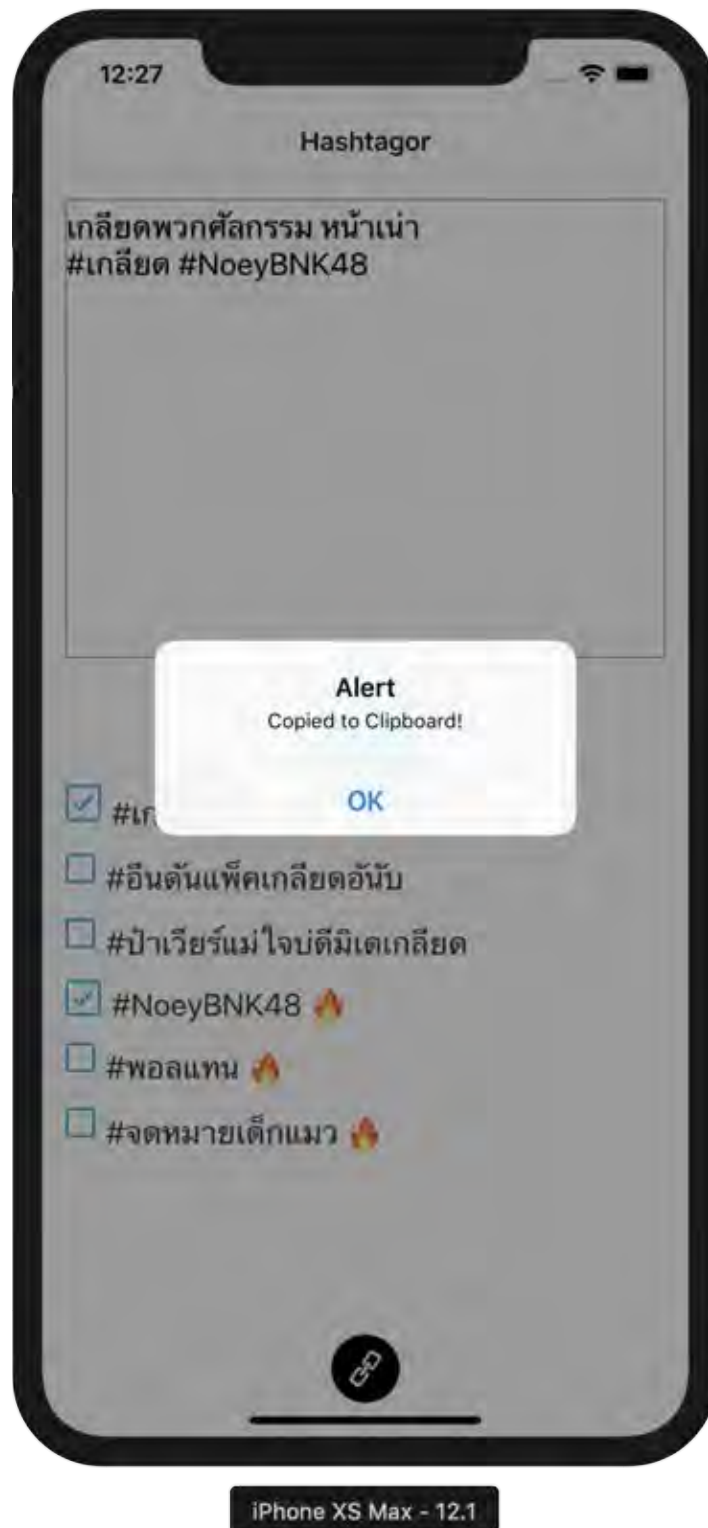


Figure 3.9 Copy to clipboard example

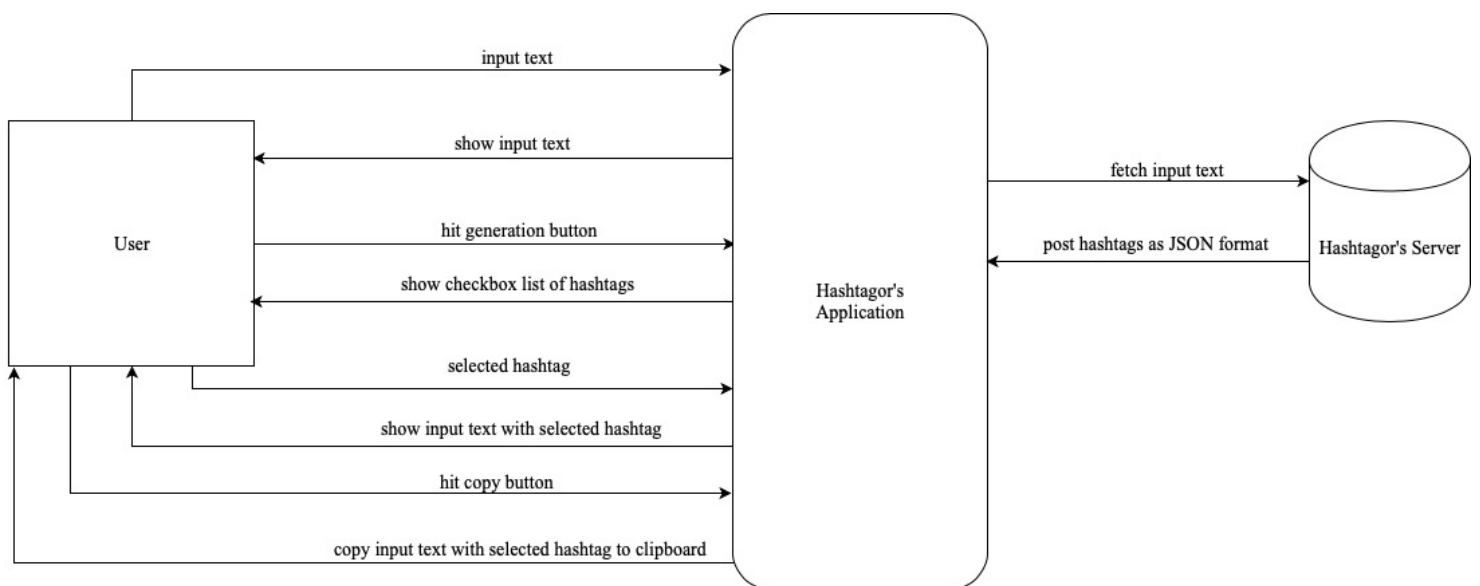
Furthermore, user can continue to use the application to discover other hashtags by editing the text in the textbox and hit operation button as usual (see Figure 3.10).



Figure 3.10 Continue using example

### 3.2 Dataflow diagram

We use state in React-native. Whenever a state is changed, the application will re-render components that are consistent with the state. So, we will set a state every time the user hit the operation button or have an interaction with our application to call our function. From the dataflow diagram in Figure 3.11, when the user gives input text, the application will show input text. When user hit the generation button and the input state is not null, the application will call a function that fetches the input from the text box and send it to the server. The application then receives hashtags in JSON format. Then, the application will transform it to a string type and show it in the form of a checkbox list. The textbox updates the input from the user with selected hashtags instantly no matter the user checks or unchecks any hashtag in the checkbox list. Finally, whenever the user hits the copy button, the application will copy the input text with selected hashtags to the clipboard.



**Figure 3.11 Hashtagor's dataflow diagram**



### 3.3 Usecase diagram

Hashtagor's usecase diagram shown as figure 3.12 and have tables to describe each case shown in table 3.1 to 3.4

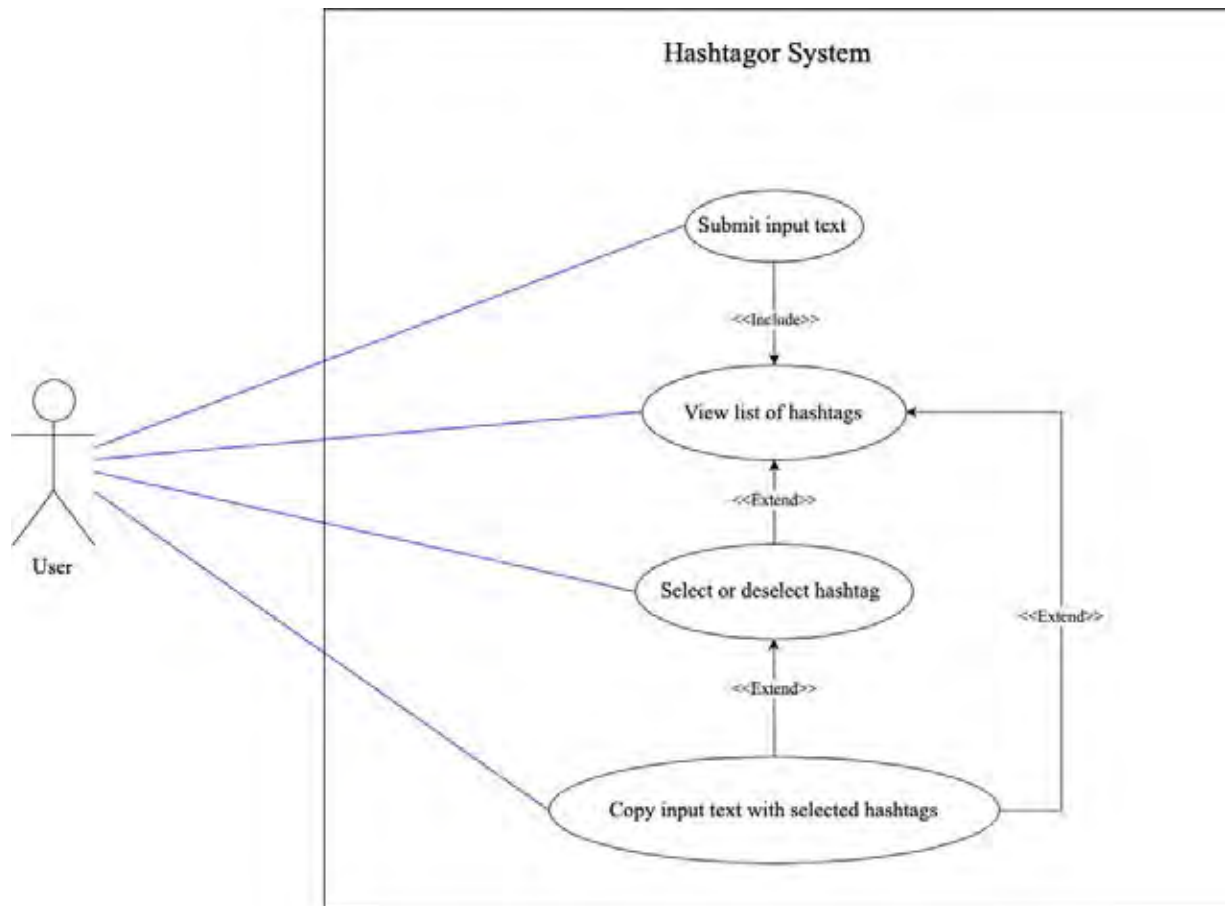


Figure 3.12 Hashtagor's usecase diagram

**Table 3.1 Usecase: Submit an input text**

Use case name	Submit an input text
Actors	User
Precondition	The user types an input text in the text box.
Postcondition	The system receives the input text.
Flow of events	1. The user clicks an operation button.

**Table 3.2 Usecase: View list of hashtags**

Use case name	View list of hashtags
Actors	User
Precondition	The user submits the input text.
Postcondition	The system represents a list of hashtags.
Flow of events	1. The user views a list of hashtags.

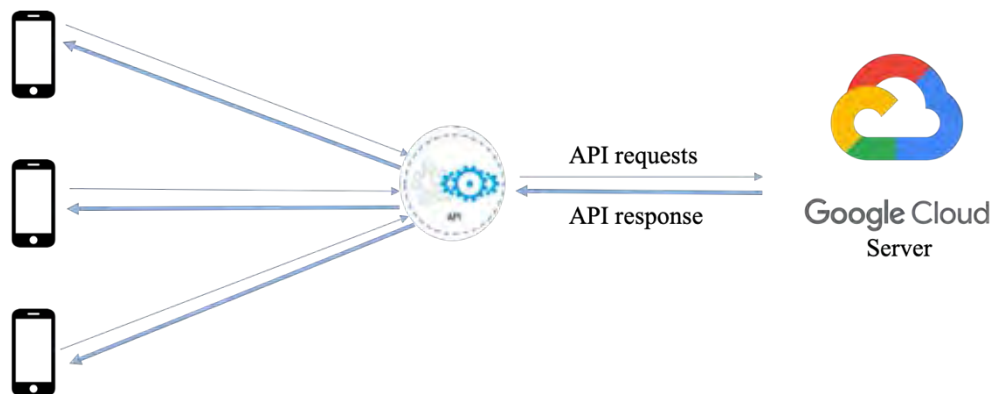
**Table 3.3 Usecase: Select or deselect hashtags**

Use case name	Select or deselect hashtags
Actors	User
Precondition	The user views a list of hashtags.
Postcondition	The system gets a list of selected hashtags.
Flow of events	1. The system updates a list of selected hashtags.

**Table 3.4 Usecase: Copy the input text with selected hashtags**

Use case name	Copy the input text with selected hashtags
Actors	User
Precondition	The user views a list of hashtags.
Postcondition	The input text and selected hashtags were copied to the clipboard.
Flow of events	1. The user copies the input text with selected hashtags.

### 3.4 Software Architecture



**Figure 3.13 Software Architecture**

We use client/server architecture as shown in Figure 3.13 because from this architecture; several users can access to server at the same time using this architecture.

### 3.5 Software Specification

#### 3.5.1 Input/Output Specification

1. Input: English or Thai (less than 300 characters)
2. Output: three generated hashtags and three most trending hashtags

#### 3.5.2 Functional Specification

1. Be able to generate proper hashtags from the input (tweet).
2. Be able to match the most three trending hashtags based on the input (tweet).
3. Be able to copy input text with selected hashtags to the clipboard.

### 3.6 Back End

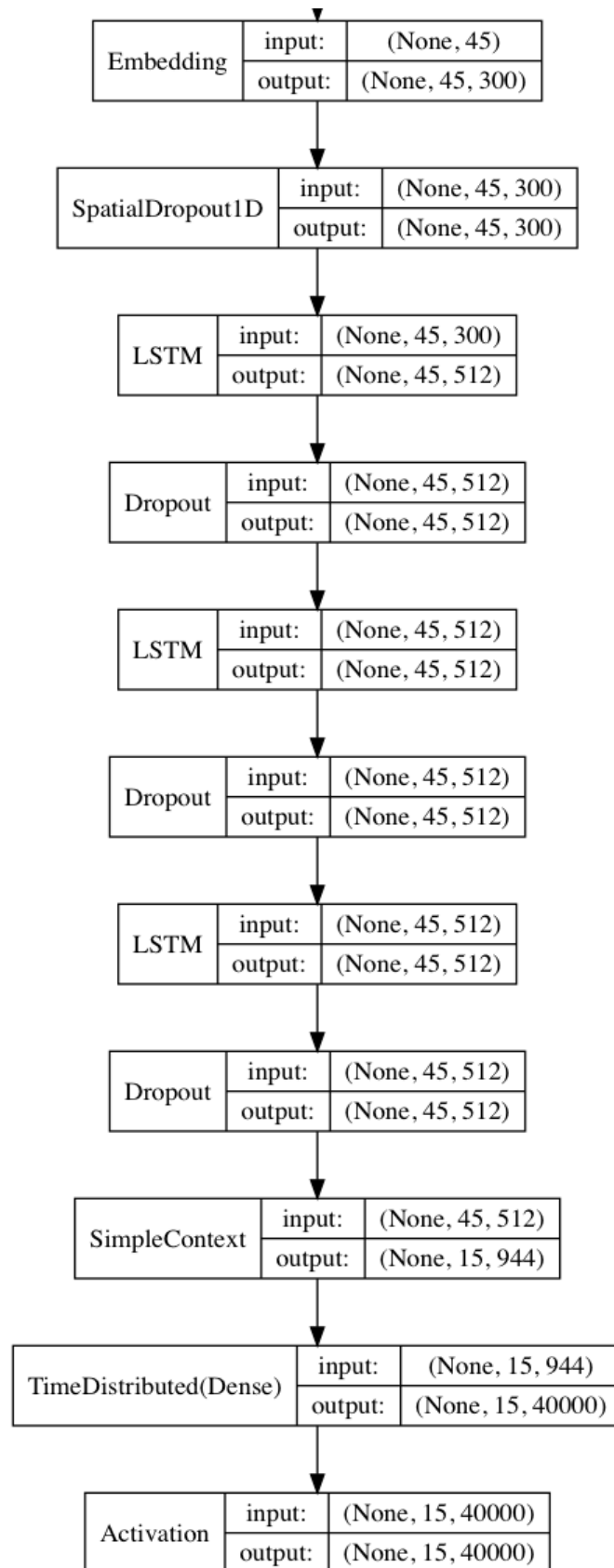
### **3.6.1 Preprocessing**

According to the project proposal, we successfully collected tweets and hashtags with Tweepy [4] from Twitter. We have cleaned and segmented (with Deepcut [3]) these data. So, in this preprocessing stage, we created a vocabulary and trained word embedding to represent a particular word as a vector using Word2Vec in gensim [7] (Skip-Gram model). After that, we converted a word to a vector and vice versa using the encoding and decoding processes.

### **3.6.2 Models**

#### **3.6.2.1 Sequence to Sequence Model**

This model is for generating hashtags function. The architecture of the Sequence to Sequence Model is the same as Figure 2.3. From the figure, we apply this model by replacing the input (description) with a representation of tweets and replacing the output (headline) with a representation of hashtags, and here is our model (see Figure 3.14 and Figure 3.15).



**Figure 3.14** Sequence to sequence model

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 45, 300)	12000000
spatial_dropout1d_2 (Spatial	(None, 45, 300)	0
lstm_1 (LSTM)	(None, 45, 512)	1665024
dropout_1 (Dropout)	(None, 45, 512)	0
lstm_2 (LSTM)	(None, 45, 512)	2099200
dropout_2 (Dropout)	(None, 45, 512)	0
lstm_3 (LSTM)	(None, 45, 512)	2099200
dropout_3 (Dropout)	(None, 45, 512)	0
simplecontext_1 (SimpleConte	(None, 15, 944)	0
time_distributed_2 (TimeDist	(None, 15, 40000)	37800000
activation_1 (Activation)	(None, 15, 40000)	0
Total params: 55,663,424		
Trainable params: 55,663,424		
Non-trainable params: 0		

Figure 3.15 Visualization of Sequence to Sequence Model for generating hashtags

## 1. Encoding

### Hyperparameters

Hyperparameters are *maxlent* (maximum of tweet length) = 30, *maxlenh* (maximum of hashtag length) = 15, *vocab\_size* (fixed vocabulary size) = 40000, *dimension* (dimensions of embedding matrix) = 300, *activation\_rnn\_size* (nodes from top LSTM layer which will be used for activation) = 40 and *rnn\_size* (nodes from LSTM used for prediction and activation) = 512

The input data (X) is made from *maxlent* tweet words followed by *<eos>* (end-of-sequence symbol) followed by *maxlenh* hashtag words followed by *<eos>*. If a tweet/hashtag is shorter than *maxlent/maxlenh*, it will be left-padded with zero value. If the complete text is longer than the maximum length, it will be right-clipped to the maximum length. Labels (Y) are the hashtag words followed by *<eos>* and right-clipped or left-padded to *maxlenh*.

### Neural Network Layer

Embedding layer: We set the *maxlen* to 45 words and train a vector representation that transform the text of 40000 vocabulary inputs into a 300-d vector.

Long Short-Term Memory (LSTM) layers: We use 512 nodes (or *rnn\_size*) for prediction and activation (according to the hyperparameter section mentioned before).

Dropout layer: To prevent overfitting.

Simple Context layer: This layer implements an attention mechanism following section 2.4.2. We use *activation\_rnn\_size* hidden units for calculating the attention weight, and another *rnn\_size - activation\_rnn\_size* for calculating the context.

Time Distributed layer: To configure the last LSTM layer before wrapping Dense layer to return sequences.

Activation layer: We use “softmax function” to obtain the probability of the neural network.

## **2. Decoding**

We use “Beam Search Decoding Algorithm” to predict the possible output words of its sample.

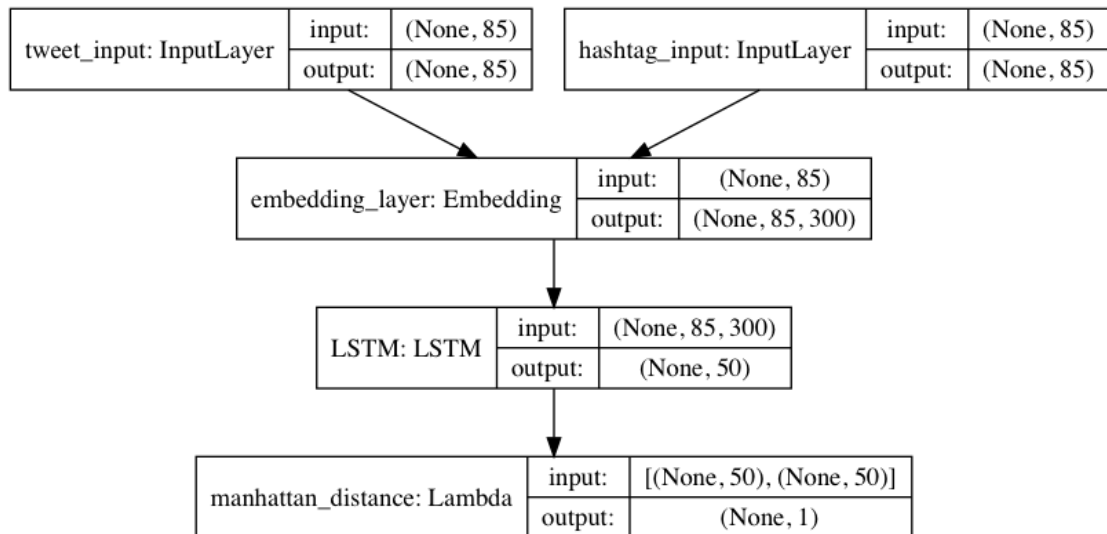
### Algorithm

While there is a sample that do not reach eos:

1. Calculate word probability for all samples (predicted by the model).
2. The total score for every sample is the sum of *-log* of word probability (sum of negative log-likelihood).
3. Find the best (lowest) scores from all possible samples.
4. Append the new words to their appropriate sample.

### 3.6.2.2 Trending Hashtags Model

We adapt the Siamese Manhattan Long Short-Term Memory Model [10] by comparing the similarity between a hashtag and its tweet instead of two question sentences. Here is our model (see Figure 3.16 and Figure 3.17).



**Figure 3.16 Trending Hashtags Model**

Layer (type)	Output Shape	Param #	Connected to
tweet_input (InputLayer)	(None, 85)	0	
hashtag_input (InputLayer)	(None, 85)	0	
embedding_layer (Embedding)	(None, 85, 300)	47475900	tweet_input[0][0] hashtag_input[0][0]
LSTM (LSTM)	(None, 50)	70200	embedding_layer[0][0] embedding_layer[1][0]
manhattan_distance (Lambda)	(None, 1)	0	LSTM[0][0] LSTM[1][0]
Total params: 47,546,100			
Trainable params: 70,200			
Non-trainable params: 47,475,900			

**Figure 3.17 Visualization of Trending Hashtags Model**



### Hyperparameters

Hyperparameters are *max\_seq\_length* (maximum of tweet and hashtag length) = 85, *dimension* (dimensions of embedding matrix) = 300 and *hidden\_node* (LSTM layer) = 50

The tweet input data (X1) is made from tweet words followed by  $\langle eos \rangle$  and the hashtag input data (X2) is made from hashtag words followed by  $\langle eos \rangle$ . If a tweet/hashtag is shorter than *max\_seq\_length*, it will be left-padded with zero value. If the complete text is longer than the maximum length, it will be right-clipped to the maximum length. Labels (Y) are integers (0's or 1's) showing whether the tweet input and the hashtag input are related (label as 1) or not (label as 0).

### Model layers

Embedding layer: We set the *max\_seq\_length* to 85 words and train a vector representation that transform the text of 158252 vocabulary inputs into a 300-d vector.

LSTM layers: We use 50 nodes (or *rnn\_size*) for prediction.

Manhattan distance (Output layer): To calculate similarity between hashtag and tweet according to this equation,

$$expo\_neg\_manhattan\_distance = \exp(-\text{sum}(\text{abs}(\text{tweet\_input} - \text{hashtag\_input})))$$

Finally, we use the score (Manhattan distance) from the model to ranking the top trending hashtag.

## CHAPTER IV

### RESULTS

This chapter shows the details of the datasets used, model testing results and application testing results.

#### 4.1 Model Testing Results

##### 4.1.1 Hashtags generating model

We scraped 283511 tweets from 14 September 2018 to 8 November 2018. We split the dataset into 233511 rows for training and 50000 rows for validation and testing. The example is shown in Figure 4.1.

	tweets	hashtags
0	อก เกือบ หัก แอบ รัก คุณ สามี สภาพ ดี ราคา ไม่...	ตาม หานิยาย
1	อก เกือบ หัก แอบ รัก คุณ สามี สภาพ ดี ราคา ไม่...	ส่ง ต่อ หนังสือ
2	อก เกือบ หัก แอบ รัก คุณ สามี สภาพ ดี ราคา ไม่...	นิยาย มือ สอง
3	อก เกือบ หัก แอบ รัก คุณ สามี สภาพ ดี ราคา ไม่...	นิยาย มือ สองสภาพ ดี
4	อก เกือบ หัก แอบ รัก คุณ สามี สภาพ ดี ราคา ไม่...	ขาย นิยาย มือ สอง
5	อก เกือบ หัก แอบ รัก คุณ สามี สภาพ ดี ราคา ไม่...	หนังสือ มือ สอง
6	ถ่าย ละคร ทั้ง วัน เลย เหนื่อย มั้ย ครับ ลู๊ ๆ...	myhusbandinlaw
7	เอา มา ฝาก แฟน ๆ ค่ะ บาง ส่วน จาก กองอก เกือบ ...	2cr
8	เอา มา ฝาก แฟน ๆ ค่ะ บาง ส่วน จาก กองอก เกือบ ...	myhusbandinlaw
9	เอา มา ฝาก แฟน ๆ ค่ะ บาง ส่วน จาก กองอก เกือบ ...	mewnitthafanclubofficial

Figure 4.1 Dataset example

We directly tested with human to see if the generated hashtags are related to the text as an input or not. Figure 4.2 and Figure 4.3 show some hashtags and scores which were generated and evaluated by the model.

```

Actual_hashtags: n5 EOS
Tweet: ธรรมชาติ ภาษา ญี่ปุ่น สนใจ เรียน ติดต่อ เบอร์ 0988 254602 lineid pasajipunhouse EOS
Predict_hashtags:
1.9943032832816243 ['เบอร์']
1.9943032832816243 ['เบอร์']
2.8488715984858572 ['กองเต้า']
2.8488715984858572 ['กองเต้า']
3.7437846083194017 ['ธรรมชาติ']
3.8403540881117806 ['ปปปปปป']
3.938620689288655 ['รักกกกกก']
4.4801741909686825 ['คริสตอาน์']
4.583457710934454 ['คุณกิ๊ก']
4.892247369571123 ['พริสส']
4.946197038334503 ['เลรอย']
5.185548070097866 ['พัตจินยอง']
5.987394157680683 ['รีวิวตา', 'ผาด']
6.86058135330677 ['รีวิวตา', 'เบอร์']
7.0615687888348475 ['ปฏิรูป']
7.078279866778757 ['สำนักข่าวนี้']
7.367070838925429 ['უნหา']
7.982797457836568 ['เบอร์', 'ธรรมชาติ']
8.512533389031887 ['รีวิวตา', 'เบอร์', 'มินิมัอม']
14.997897375957109 ['เบอร์', 'ธรรมชาติ', 'พริสส']

```

Figure 4.2 Hashtags and scores1

```

Actual_hashtags: jaehyun 1920 EOS
Tweet: plsrt แบบ แจชยอน fav ใน ลิงค์ มัด จำ ได้ สินค้า nct EOS
Predict_hashtags:
0.7405085316859186  ['ได้']
0.7405085316859186  ['ได้']
3.740449280710891  ['ล่าลูกทรี']
3.740449280710891  ['ล่าลูกทรี']
3.931133395526558  ['90thanniversary']
4.625000583939254  ['ใน']
4.7000002259155735  ['เอ็นลิทีเซ็นเมนแจชยอน', 'gotoon', 'ห้ห้ห้ห้ห้ห้']
4.842338665330317  ['เบอร์']
4.842338665330317  ['เบอร์']
4.900552853941917  ['010203070809']
5.045401707291603  ['น้องนารี']
5.065231718181167  ['พริต']
5.116938234659756  ['ปปปปปป']
5.273757764836773  ['กองเค']
6.917905093730951  ['อ็พชอพ']
7.124473571777344  ['เอ็นลิทีเซ็นเมนแจชยอน']
7.728670838070684  ['พ็ชุนพ็จุน']
8.867442071437836  ['เพอ']
9.327085308730602  ['เอ็นลิทีเซ็นเมนแจชยอน', 'gotoon', 'มีนก็อชยอก']
14.840596865862608  ['ได้', 'shower']

```

Figure 4.3 Hashtags and scores2

#### 4.1.2 Trending hashtags model

We get a dataset 590827 pairs of tweets and hashtags, and we divide tweets and trending hashtags pairs into two inputs and one output as a label of each pair whether it is trending or not, then there 0's and 1's equal to 380392 rows and 210435 rows respectively. The data example is shown in Figure 4.4.

	tweets	hashtags	is_trending
20	[ถ่าย, ละคร, ทั้ง, วัน, เลย, เหนื่อย, มั้ย, คร...	[myhusbandinlaw]	0
21	[ถ่าย, ละคร, ทั้ง, วัน, เลย, เหนื่อย, มั้ย, คร...	[หมาก]	0
22	[ถ่าย, ละคร, ทั้ง, วัน, เลย, เหนื่อย, มั้ย, คร...	[หมากปริญ]	0
23	[ถ่าย, ละคร, ทั้ง, วัน, เลย, เหนื่อย, มั้ย, คร...	[prinfanclub]	0
24	[พี่เขียร, รรร, จุ้, ๆ, ๆ, ๆ]	[อก, เกือบ, หัก, แอบ, รัก, คุณ, สามี]	1
25	[เอา, มา, ฝาก, แฟน, ๆ, ค่ะ, บาง, ส่วน, จาก, กอ...	[อก, เกือบ, หัก, แอบ, รัก, คุณ, สามี]	1
26	[เอา, มา, ฝาก, แฟน, ๆ, ค่ะ, บาง, ส่วน, จาก, กอ...	[2cr]	0
27	[เอา, มา, ฝาก, แฟน, ๆ, ค่ะ, บาง, ส่วน, จาก, กอ...	[myhusbandinlaw]	0
28	[เอา, มา, ฝาก, แฟน, ๆ, ค่ะ, บาง, ส่วน, จาก, กอ...	[mewnitthafanclubofficial]	0
29	[เอา, มา, ฝาก, แฟน, ๆ, ค่ะ, บาง, ส่วน, จาก, กอ...	[อก, เกือบ, หัก, แอบ, รัก, คุณ, สามี]	1

Figure 4.4 Data example

We train on 400000 samples, validate on 100000 samples and test on 90827 samples.

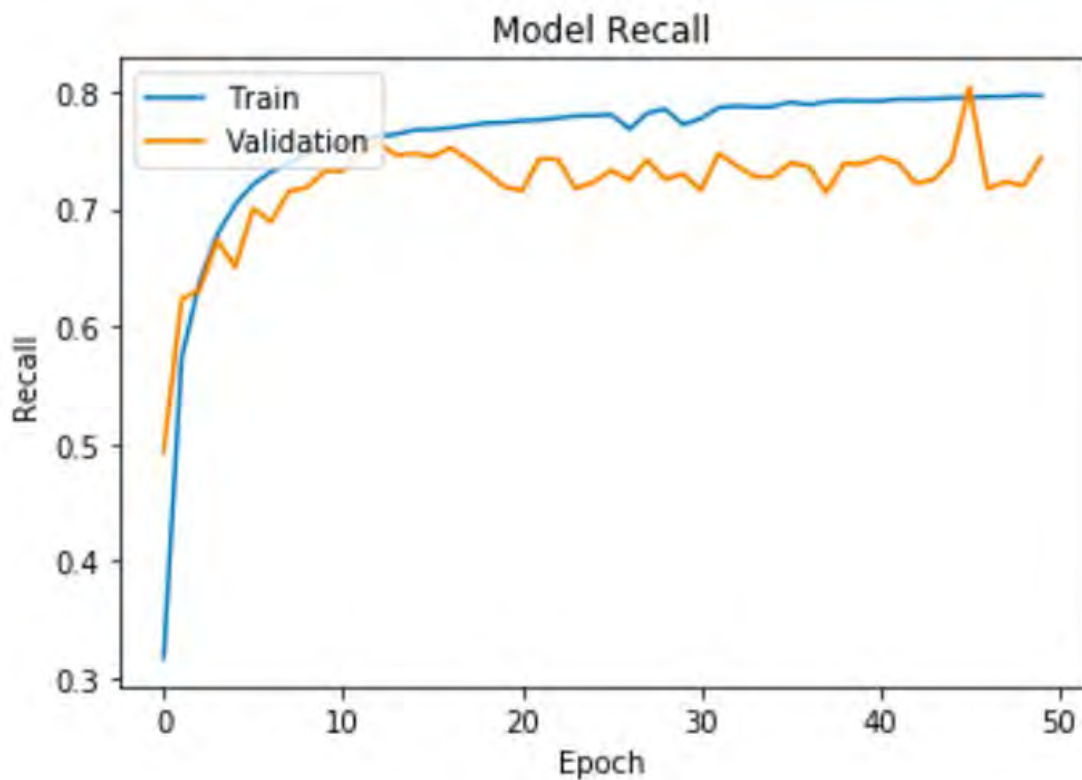
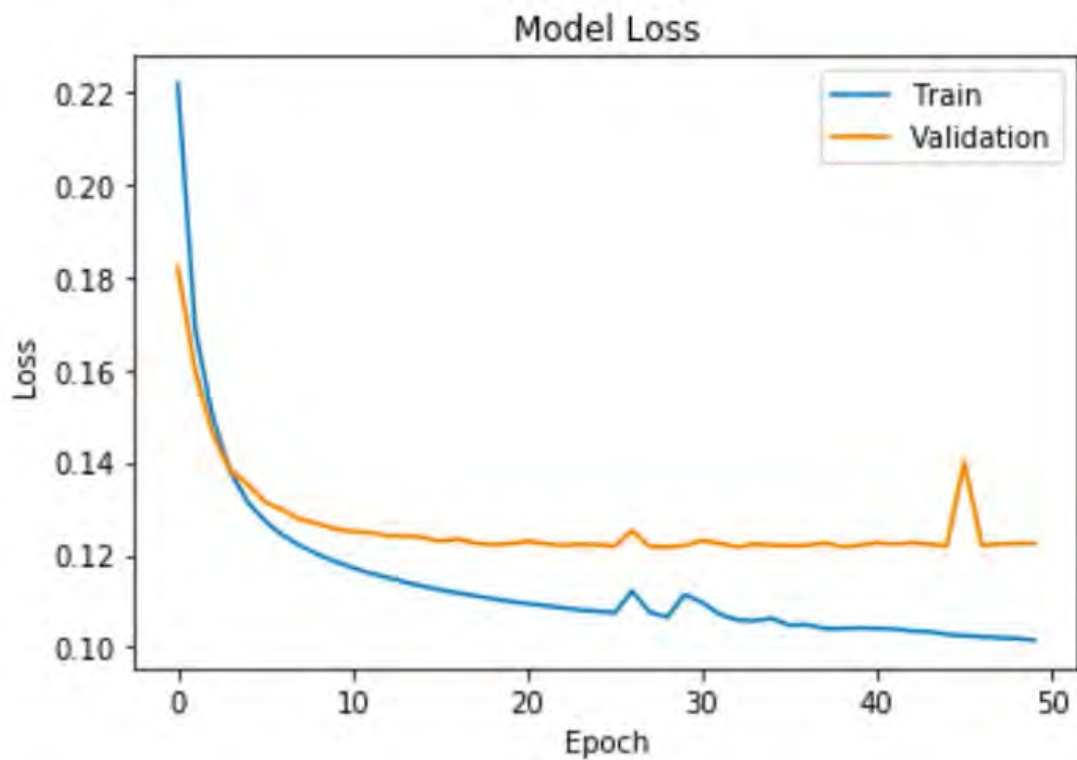


Figure 4.5 Model recall



**Figure 4.6 Model loss**

Figure 4.5 and Figure 4.6 show the results on the train and validation sets after training 50 epochs. We choose the final model at epoch 32 which the validation sets reach the lowest loss score (0.12168) and achieve recall rate at 76.51%.

The final model achieves recall rate at 55.47% on the test sets.

## 4.2 Application Testing Results

The results show that the mobile application can receive input from users' mobile and successfully get the results as hashtags as desired (see Figure 4.7 and Figure 4.8).

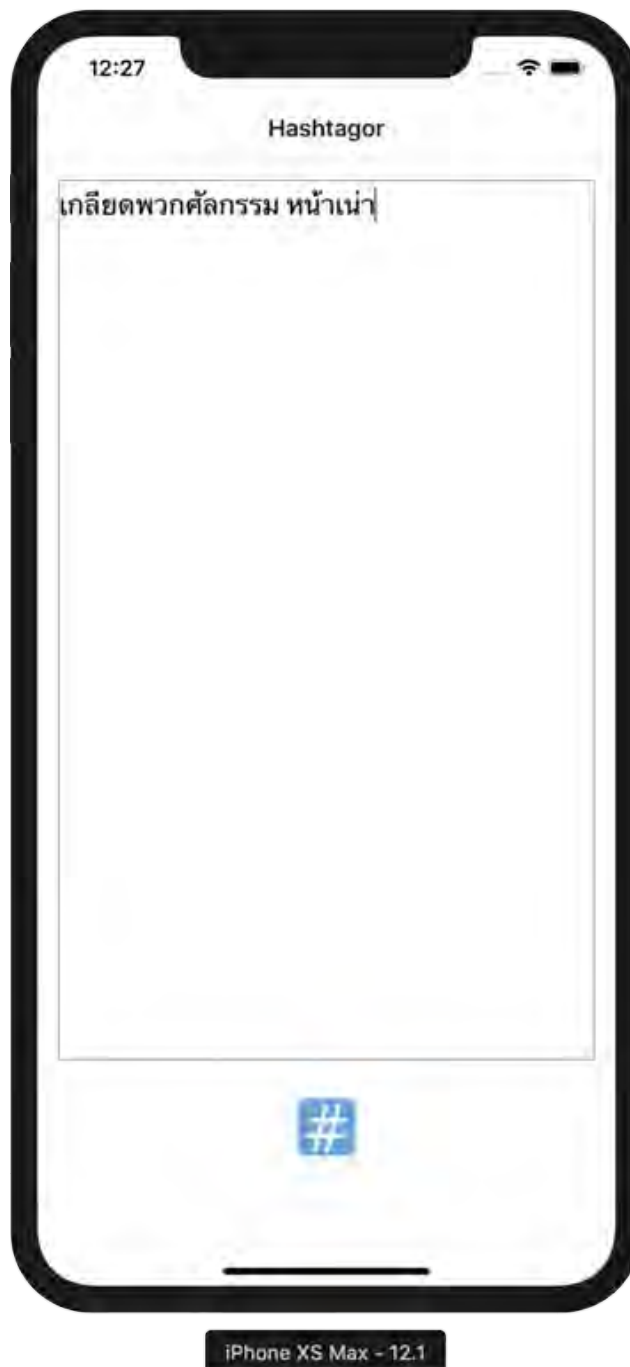


Figure 4.7 Before hit operation button example.



Figure 4.8 Result after hit Operation button example



Moreover, the application can copy text with selected hashtags to the clipboard (see Figure 4.9).

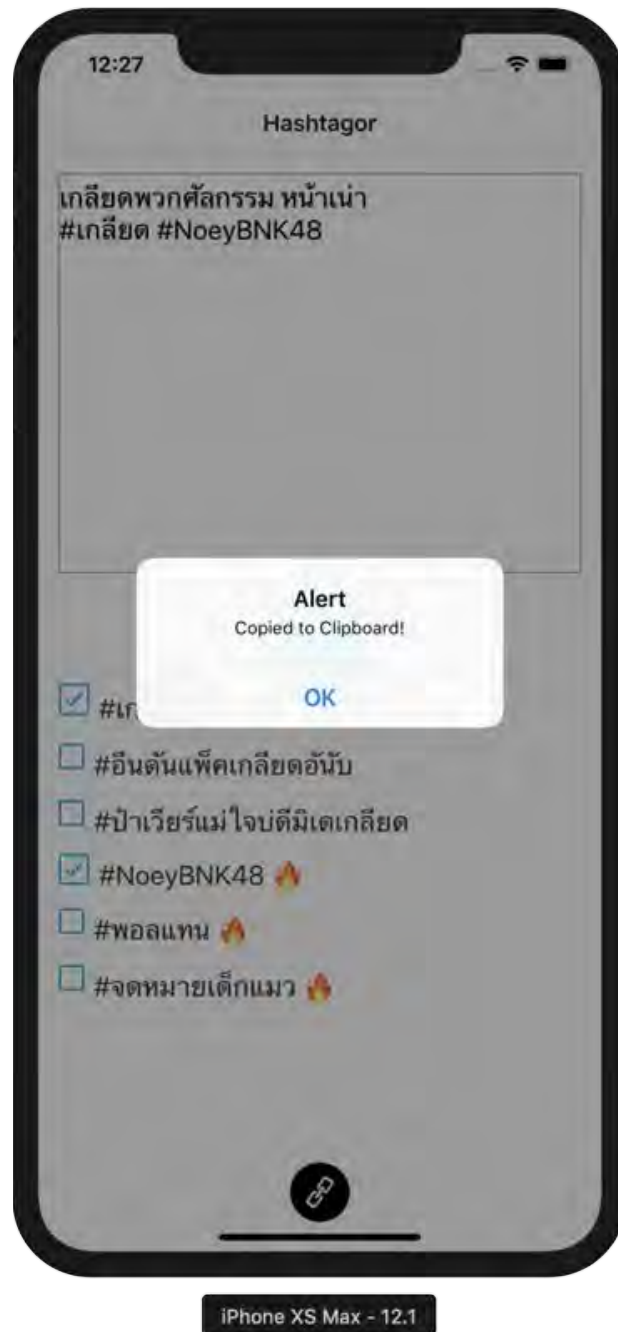


Figure 4.9 Result after hit Copy button example

## **CHAPTER V**

### **CONCLUSION**

This chapter has the conclusions of this report and provides a suggestion for further study on this topic.

#### **5.1 Conclusion**

We have implemented a mobile application on the iOS platform using the React Native framework. Our application can recommend new hashtags and trending hashtags that are related to the given tweets to a user. From the results of the mobile interface application, everything is working as expected. However, the results from the trending hashtags model are not as good as we expect because of the limitation of computation cost and time. So, the models need to be trained for a more extended amount of time to get better parameters or weights to achieve the better output. The application also works as we expect but the only share button to Twitter gets stuck which we aim to fix it in the future.

#### **5.2 Suggestion**

1. We have found that Expo does not support the share feature. The future update should replace this library with other libraries that support iOS 11 and upper.
2. Request for an apple developer account and deploy this project on apple store so the user can download the application from the app store.
3. The current fixed vocabulary size on Sequence to Sequence model for generating hashtags is too small. If we increase the fixed vocabulary size, the model may achieve a better recall.
4. Performance of the word embedding is crucial in the trending hashtags model. We need to prepare more data to get better embeddings. One way to achieve such performance is to add more sentences to the training set.

5. Expo has conflicts with React Native Link. Further improvement may require switching from using Expo to using Native language for iOS.

## REFERENCES

- [1] Simon Kemp. (2018). Digital in 2018: World's Internet Users Pass The 4 Billion Mark. Retrieve from <https://wearesocial.com/blog/2018/01/global-digital-report-2018> [14 August 2018]
- [2] Dominique Jackson. (2018). How to Find the Best Twitter Hashtags. Retrieve from <https://sproutsocial.com/insights/twitter-hashtags/> [29 August 2018]
- [3] Rakpong Kittinaradorn. (2018). Deepcut. Retrieve from <https://github.com/rkcosmos/deepcut> [30 August 2018]
- [4] Aaron Hill and Joshua Roesslein. (2018). Tweepy: Twitter for Python! Retrieve from <https://github.com/tweepy/tweepy> [1 September 2018]
- [5] pandas-dev. (2018). Pandas: powerful Python data analysis toolkit. Retrieve from <https://github.com/pandas-dev/pandas> [1 September 2018]
- [6] TensorFlow. (2018). API Document. Retrieve from [https://www.tensorflow.org/api\\_docs/](https://www.tensorflow.org/api_docs/) [1 September 2018]
- [7] Gensim. (2018). Word2Vec embeddings. Retrieve from <https://radimrehurek.com/gensim/models/word2vec.html> [15 September 2018]
- [8] Don Vetal. (2018). Skip-Gram Architecture overview. Retrieve from <https://www.districtdatalabs.com/nlp-research-lab-part-2-skip-gram-architecture-overview> [19 April 2018].
- [9] Niklas Donges. (2018). Long-Short Term Memory. Retrieve from <https://machinelearning-blog.com/2018/02/21/recurrent-neural-networks> [19 April 2018].
- [10] Jonas Mueller and Aditya Thyagarajan. (2016). Siamese recurrent architectures for learning sentence similarity. In Thirtieth AAAI Conference on Artificial Intelligence.
- [11] Elior Cohen. (2017). How to predict Quora Question Pairs using Siamese Manhattan LSTM. Retrieve from <https://medium.com/mlreview/implementing-malstm-on-kaggles-quora-question-pairs-competition-8b31b0b16a07> [14 April 2019]

- [12] Konstantin Lopyrev. (2015). Generating News Headlines with Recurrent Neural Networks. arXiv/1512.01712.
- [13] Facebook Open Source. (2019). React Native. Retrieve from <https://facebook.github.io/react-native> [23 January 2019]
- [14] Facebook Open Source. (2019). React. Retrieve from <https://reactjs.org> [23 January 2019]
- [15] Adhithi Ravichandran. (2018). Props and State in React Native explained in Simple English. Retrieve from <https://codeburst.io/props-and-state-in-react-native-explained-in-simple-english-8ea73b1d224e> [7 April 2019]
- [16] Bartosz Szczeciński. (2017). Understanding React-Component Life-Cycle. Retrieve from <https://medium.com/@baphemot/understanding-reactjs-component-life-cycle-823a640b3e8d> [7 April 2019]
- [17] React Navigation. (2019). React Navigation. Retrieve from <https://reactnavigation.org> [23 January 2019]
- [18] Expo. (2019). Expo SDK. Retrieve from <https://expo.io> [14 April 2019]
- [19] Richard N. Taylor, Nenad Medvidovic, and Eric M. Dashofy. (2008) Software Architecture: Foundations, Theory, and Practice. John Wiley & Sons, Inc. Reprinted with permission.

## **APPENDICES**

**APPENDIX A**  
**The Project Proposal of Course 2301399 Project Proposal**  
**Academic Year 2061**

**แบบเสนอหัวข้อโครงการ รายวิชา 2301399 Project Proposal**  
**ปีการศึกษา 2561**

ชื่อโครงการ (ภาษาไทย)	แฮชแท็กเกอร์
ชื่อโครงการ (ภาษาอังกฤษ)	Hashtagor
หัวหน้าภาควิชา	ศ. ดร. กฤษณะ เนียมมณี
อาจารย์ที่ปรึกษา	ผศ. ดร. จิตยา หวานวารี
ผู้พัฒนา	นายชินนทร์ชัย หาญเมือง นางสาวขวัญชนก ศรีสมพงษ์ สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

---

**หลักการและเหตุผล**

สำหรับมนุษย์แล้วการเข้าสังคมถือเป็นเรื่องที่ไม่ได้ เพราะมนุษย์ก็เป็นสัตว์สังคมชนิดหนึ่ง ที่ต้องการการติดต่อสัมพันธ์ซึ่งกันและกัน เครื่องมือที่ใช้ในการสื่อสารก็พัฒนามาเรื่อยๆ ตั้งแต่ยุค นักพิราบสื่อสารจดหมายโทรสารจนถึงยุคที่เป็นเครือข่ายสังคมออนไลน์ที่ได้เข้ามามีบทบาทสำคัญในการติดต่อสื่อสารเป็นอย่างมากโดยเฉพาะในประเทศไทยเป็นประเทศที่มีเวลาการใช้งานโปรแกรมประยุกต์เพื่อการสื่อสารต่อวันมากที่สุดในโลก [1] เช่น Facebook Twitter และ Instagram ซึ่งเป็นโปรแกรมที่ใช้ติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ต ที่เป็นที่ยอมรับในหลากหลายช่วงอายุ

ในปัจจุบัน สังคมมีความซับซ้อนมากขึ้น นอกจากข้อมูลจากสื่อต่างๆแล้ว ในตอนนี้ผู้ใช้ทุกคนสามารถเป็นผู้ให้ข้อมูลข่าวสารได้ ทำให้ข้อมูลในสารบบหลากหลายและปะปนกัน จึงเป็นเรื่องยากที่จะหากลุ่มสังคมที่มีความสนใจประเด็นเดียวกัน จึงเกิดสิ่งที่เรียกว่า Hashtag หรือเครื่องหมาย # เป็นตัวช่วยหากลุ่มของสังคมที่สนใจในหัวข้อนั้นๆหรือสามารถตรวจสอบได้ว่าหัวข้อใดที่กำลังเป็นที่นิยมอยู่ ณ ขณะนั้น ตัวอย่างการใช้ Hashtag ในข้อความ (Tweet) บน Twitter เช่น “ลองไหว้ชั้นสวยๆสักทีสิ ชั้นอาจจะให้อีกสัก 2 3 ล้าน #แรงเงา” ซึ่งผู้ใช้นบน Twitter สามารถ

คลิกที่ # แรงเงา เพื่อเข้าไปดูข้อความหรือ Tweet ของคนอื่นที่ใช้ Hashtag นี้เพื่อดูบทความของผู้อื่นที่กล่าวถึงในหัวข้อตาม Hashtag เดียวกันนี้

แต่การจะกำหนด Hashtag ขึ้นมาในบางครั้งไม่ใช่เรื่องง่าย เนื่องจากผู้ใช้อาจกำหนด Hashtag ได้ไม่เหมาะสมกับเนื้อหาหรือคำใน Hashtag ที่คิดขึ้นมานั้นไม่ได้เป็นที่นิยมในสังคม ไม่สามารถเชื่อมโยงถึงประเด็นที่กำลังเป็นที่นิยมอยู่ในขณะนั้นได้ [2] ซึ่งปัญหาที่ตามมาก็คือ ผู้ใช้ไม่สามารถหาเนื้อหาหรือบทความที่กล่าวถึงเรื่องเดียวกันนี้ได้เนื่องจากการตั้ง Hashtag ที่ไม่เหมือนกัน แต่ถ้าหากผู้ใช้รู้ความนิยมหรือกระแสที่กำลังแรงอยู่ใน Twitter ก็อาจจะอิงกระแสนั้นมาคิดเป็น Hashtag ใหม่ได้ซึ่งอาจทำให้ Hashtag ที่คิดขึ้นมาใหม่นั้นเป็นกระแสได้

จากปัญหานี้จึงเกิดการพัฒนาระบบโปรแกรมประยุกต์เพื่อแนะนำ Hashtags โดยใช้ข้อมูลจาก Twitter มาเรียนรู้เพื่อประมวลผล 2 ข้อดังนี้

1. การสังเคราะห์ Hashtags ขึ้นมาจากข้อความใน Tweet ของผู้ใช้
2. การเลือก Hashtags ที่กำลังอยู่ในความนิยมของประเทศไทยที่เหมาะสมกับข้อความใน Tweet ของผู้ใช้

### วัตถุประสงค์

1. สร้างโปรแกรมประยุกต์ที่สามารถสังเคราะห์ Hashtag ใหม่และแนะนำหัวข้อที่อยู่ในความนิยมโดยอิงจากข้อความใน Tweet บน Twitter ที่เป็นข้อความภาษาไทยใน Twitter
2. โปรแกรมจะต้องใช้งานง่าย ไม่ซับซ้อน รองรับ iOS และสามารถแบ่งปัน Hashtag นี้ไปยัง Twitter และคัดลอกไปยังคลิปปอร์ดได้

### ขอบเขตของโครงการ

1. รองรับเฉพาะระบบปฏิบัติการ iOS
2. สามารถแบ่งปันข้อความกับ Twitter รวมทั้งโปรแกรมประยุกต์เพื่อการสื่อสารอื่นๆได้
3. ใช้ข้อความใน Tweet ที่ติด Trending ประเทศไทยบน Twitter เท่านั้น
4. ใช้ข้อความที่เป็นภาษาไทยหรือภาษาอังกฤษเท่านั้น



## วิธีการดำเนินงาน

1. ศึกษาความรู้เกี่ยวกับ การตัดคำ [3] API ของ Twitter [4] ตารางข้อมูลใน Python [5] การใช้โครงข่ายประสาทเทียม [6] และ การแปลงคำเป็นเวกเตอร์ด้วย Gensim [7]
2. จัดเตรียมข้อมูล

2.1 เก็บ Hashtag หรือ Topic จาก Trending ของ Twitter โดย Hashtag คือข้อความที่มีเครื่องหมาย # นำหน้าข้อความและ หัวข้อ (topic) คือข้อความที่ไม่มี # นำหน้าข้อความ ซึ่ง Tweepy API [4] สามารถดึงทั้ง Hashtag และหัวข้อได้ โดยเก็บหัวข้อและ Hashtag นี้เพื่อที่จะนำไปใช้กับ Tweepy API [4] ในการค้นหาข้อความจากหัวข้อหรือ Hashtag เพื่อที่จะนำคำในข้อความเหล่านี้แปลงเป็น Vector โดยใช้ [7] และจึงนำ Vector เหล่านี้ไปใช้ในแบบจำลองเพื่อสังเคราะห์ Hashtag ใหม่ต่อไป

ทำการเก็บข้อมูลตั้งแต่วันที่ 14 กันยายน 2561 โดยมีเป้าหมายเก็บให้มากกว่า 2000 Hashtag ซึ่ง ณ วันที่ 8 พฤศจิกายน 2561 ได้ 2225 Hashtag/Topic ซึ่งเป็นจำนวนที่อยู่ในช่วงที่ต้องการแล้ว



```

#ดาวสว่าง
#วันแรกโลก
#MyYouthJinyoungDay
THANK YOU THAILAND
#SOTUSEncoreFanMeetingInTaipei
#เดอแปงที่ไทหนเมื่อไหร่ก็อวย
#ศรราม
#peaceday
#ItsDJHY0day
#คึกคึกนราเธอร์
#DancingHigh
#Japan
#NetflixTH
#ปรมาจารย์ลัทธิมาร
#NANN
#ช่อง8
#kcon2018inthailand
finished
598
finished
598

```

#askmetop3	#IntoTheLigh	#TheJudgem	#อกเกือบหัก	#AppleEvent	iPhone XR	Apple Watch
14/9/18	14/9/18	14/9/18	14/9/18	14/9/18	14/9/18	14/9/18

ตัวอย่างข้อมูลที่ได้จากการใช้ API ของ Twitter

2.2 ตัว API ของ Tweepy [4] นั้นสามารถดึง Tweet หรือข้อความโดยค้นหาจาก หัวข้อ และ วันที่ ตามที่ได้เก็บมาจากขั้นที่ 2.1

2018-09-22 07:18:39  
 แค่อี้อ เรื่องก็ฟินแล้ว #อกเกือบหักแอบรักคุณสามี <https://t.co/NpFerZ0UXm>  
 2018-09-22 05:50:12  
 แคล่ตั้งตัวละครอื่นทยอยมาแล้ว  
 หมอเจียบ แพร แพรว คณิตกุล  
 #อกเกือบหักแอบรักคุณสามี <https://t.co/TVsVDe009q>  
 2018-09-22 05:49:51  
 #ด้วยแรงอธิษฐาน เห็นมีวใน เรื่องนี้แล้วรอ #อกเกือบหักแอบรักคุณสามี เลย รู้สึกว่าหลังๆมันได้เล่นหลายบทบาทมากขึ้น ชอบนะ ชอบกว่าตอนเล่นเป็นรสาในคุณชายปวารรุจอีกก มีความตังาม  
 2018-09-22 05:24:52  
 ซอบบบบ #อกเกือบหักแอบรักคุณสามี <https://t.co/CpnUXCR9rW>  
 2018-09-22 05:08:36  
 คู่มเพื่อนเมยปะนิ #อกเกือบหักแอบรักคุณสามี <https://t.co/fGAhwB7Fsy>  
 2018-09-22 04:16:45  
 #อกเกือบหักแอบรักคุณสามี เปิดแคสต์เพิ่มแล้วนะคะ ฟอรัยคก็มีแพรวก็มาได้ 😊 อ  
 📌 preawkanitkul <https://t.co/wl0hGveBqI>  
 2018-09-22 01:32:11  
 อยากรูแล้วว จริงๆได้อ่านนิยายเรื่องนี้เพราะเห็นรูปพืดตั้งของหมากมีวเลยไปชื้อนิยายมาอ่าน ละก็รู้สึกว่าเป็นเด็กเซียร์นี้หมากได้อ่าา มีวต้องเป็นน้องเม

ตัวอย่าง Tweet ที่ค้นหาด้วยคำว่า #อกเกือบหักแอบรักคุณสามี ของวันที่ 22 กันยายน 2561

2.3 ทำความสะอาดข้อความโดยการตัดอรรถที่ไม่ใช่ เช่น เครื่องหมายพิเศษ URL และ Emoji ซึ่งเป็นผลมาจากขอบเขตของงานที่จะนำข้อความที่เป็นเฉพาะภาษาไทย ภาษาอังกฤษ และตัวเลขมาวิเคราะห์เท่านั้น

แค่อี้อ เรื่องก็ฟินแล้ว อกเกือบหักแอบรักคุณสามี  
 แคล่ตั้งตัวละครอื่นทยอยมาแล้วหมอเจียบ แพร แพรว คณิตกุลอกเกือบหักแอบรักคุณสามี  
 ด้วยแรงอธิษฐาน เห็นมีวใน เรื่องนี้แล้วรอ อกเกือบหักแอบรักคุณสามี เลย รู้สึกว่าหลังๆมันได้เล่นหลายบทบาทมากขึ้น ชอบนะ ชอบกว่าตอนเล่นเป็นรสาในคุณชายปวารรุจอีกก มีความตังาม  
 ซอบบบบ อกเกือบหักแอบรักคุณสามี  
 คู่มเพื่อนเมยปะนิ อกเกือบหักแอบรักคุณสามี  
 อกเกือบหักแอบรักคุณสามี เปิดแคสต์เพิ่มแล้วนะคะ ฟอรัยคก็มีแพรวก็มาได้ อ preawkanitkul  
 อยากรูแล้วว จริงๆได้อ่านนิยายเรื่องนี้เพราะเห็นรูปพืดตั้งของหมากมีวเลยไปชื้อนิยายมาอ่าน ละก็รู้สึกว่าเป็นเด็กเซียร์นี้หมากได้อ่าา มีวต้องเป็นน้องเมยที่น่ารักแน่ๆ รอคอยๆ อกเกือบหักแอบรักคุณสามี  
 เด็กเซียร์ แพล่คนี่มากกก อกเกือบหักแอบรักคุณสามี หมากปรัย ชอบคุณภาพหมากนะคะ  
 รูปพืดตั้งของ หมอเจียบ amp แพร ละครเรื่อง อกเกือบหักแอบรักคุณสามี myhusbandinlaw jeablalana jeablalana หมอเจียบ เจียบลลนา cr

ตัวอย่างข้อความใน Tweet ที่ถูกทำความสะอาดแล้ว

```

['askmetop3',
 'intothelightwith9x9',
 'thejudgement',
 'อกเกือบหักแอบรักคุณสามี',
 'appleevent',
 'iphone xr',
 'apple watch series 4',
 'aung san suu kyi',
 'world economic forum',
 'samsung',
 'mark',
 'akb48',
 'hyuna',
 'nobodylikeyou',
 'นาคิ2',
 'linetodayพาดหัวชิงทรัพย์',
 'we young',
 'gtfo',
 'thepredator',

```

ตัวอย่าง Topics ที่ถูกทำความสะอาดแล้ว

## 2.4 ตัดแบ่งคำโดยใช้คลังโปรแกรม Deepcut [3]

```
In [146]: tweet_token = deepcut.tokenize("ตุ้มเพื่อนเมยปะนิ อกเกือบหักแอบรักคุณสามี")
```

```
In [148]: tweet_token
```

```
Out[148]: ['ตุ้ม',
           'เพื่อน',
           'เมย',
           'ปะ',
           'นิ',
           'อก',
           'เกือบ',
           'หัก',
           'แอบ',
           'รัก',
           'คุณ',
           'สามี']
```

ตัวอย่างการตัดคำว่า “ตุ้มเพื่อนเมยปะนิ อกเกือบหักแอบรักคุณสามี”

2.5 คุณสติดิของคำเพื่อวิเคราะห์หาคำที่อาจไม่จำเป็น (stopword) หรือดูว่าคำไหนมีความสำคัญกับหัวข้อนั้นๆ

	Freq
รัก	139
แอบ	106
เกือบ	106
หัก	106
อก	101
คุณ	85
สามี	84
ๆ	45
มา	32
นา	29
แล้ว	26
นี้	23
คุณสามี	23
ไม่	22
ได้	21
เป็น	19
ก็	19
นะ	18
มาก	18
ละคร	17
ทีม	16
เมย์	16
ไป	16
รูป	15
จะ	15
ภาพ	15
...	15

ตัวอย่างความถี่ของการใช้คำต่างๆใน Tweet ที่ได้จากหัวข้อ  
 “#อกเกือบหักแอบรักคุณสามี” ทั้งหมด 100 ข้อความ

### 3. สังเคราะห์ข้อมูล

โดยสร้างตัวแบบจำลองเพื่อใช้ในการเลือก Hashtag ยอดนิยมและสร้าง Hashtag ใหม่ การเรียนรู้ของเครื่อง (Machine learning) กำหนด

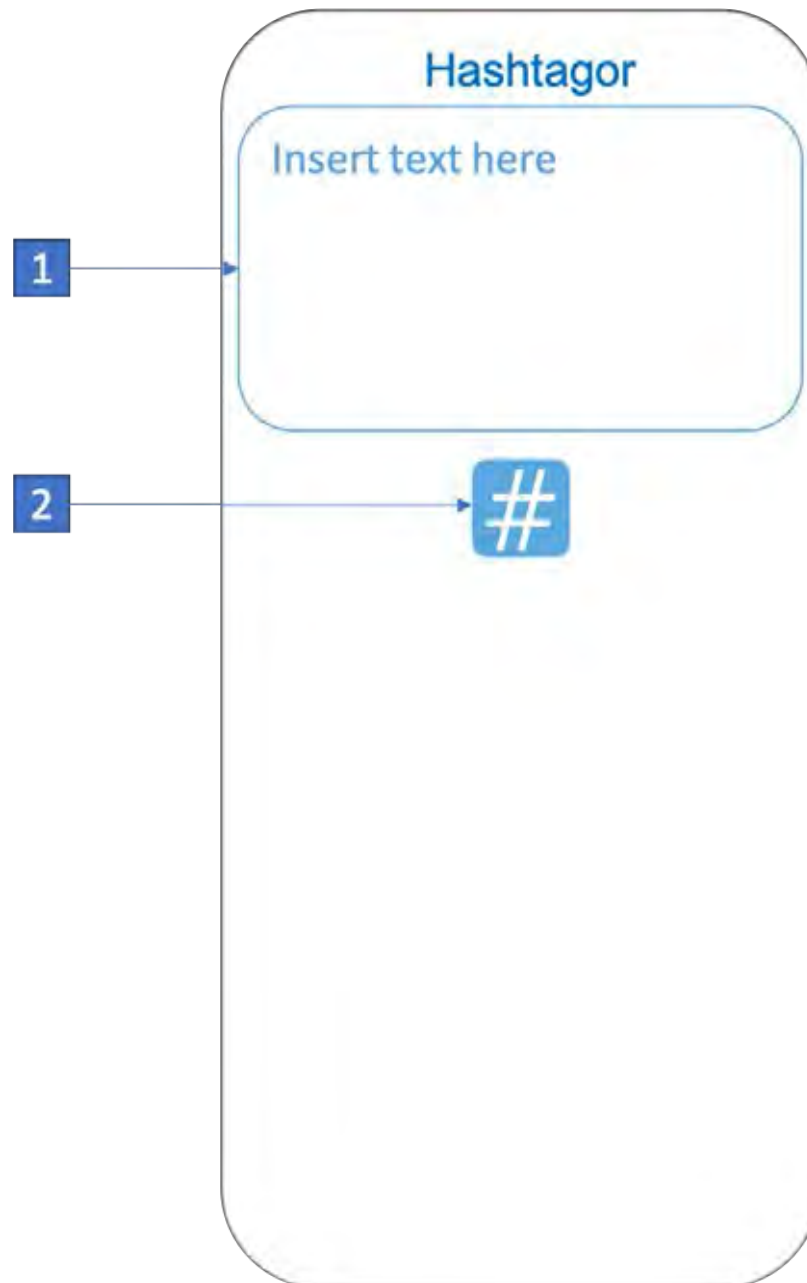
- 3.1. ข้อมูลที่ใช้สำหรับการฝึกสอนคือข้อความใน Tweet เป็นข้อมูลขาเข้า (input) และหัวข้อหรือ Hashtag ของ Tweet นั้นๆเป็นข้อมูลขาออก (output)
  - 3.2. ใช้เวกเตอร์แทนคำกับคำในข้อความทั้งข้อมูลขาเข้าและข้อมูลขาออก
  - 3.3. นำเวกเตอร์แทนคำไปใช้กับตัวแบบ Sequence-to-Sequence
  - 3.4. ทดสอบตัวแบบเพื่อดูค่าความถูกต้อง (Accuracy) ในการทำนายหัวข้อ จากข้อความใน tweet
4. ศึกษาความรู้เกี่ยวกับการพัฒนาโปรแกรมประยุกต์บนระบบปฏิบัติการ iOS
  5. พัฒนาโปรแกรมประยุกต์ โดยมีแบบจำลอง User interface ดังนี้
    - 5.1 หน้าต้อนรับ  
แสดงโลโก้และชื่อโปรแกรมประยุกต์ขณะเข้าสู่โปรแกรมประยุกต์



## 5.2 หน้าดำเนินการ

### 5.2.1 หน้ารับข้อมูลนำเข้า

ประกอบด้วย กล่องรับข้อความ (1) และปุ่มประมวลผล (2)

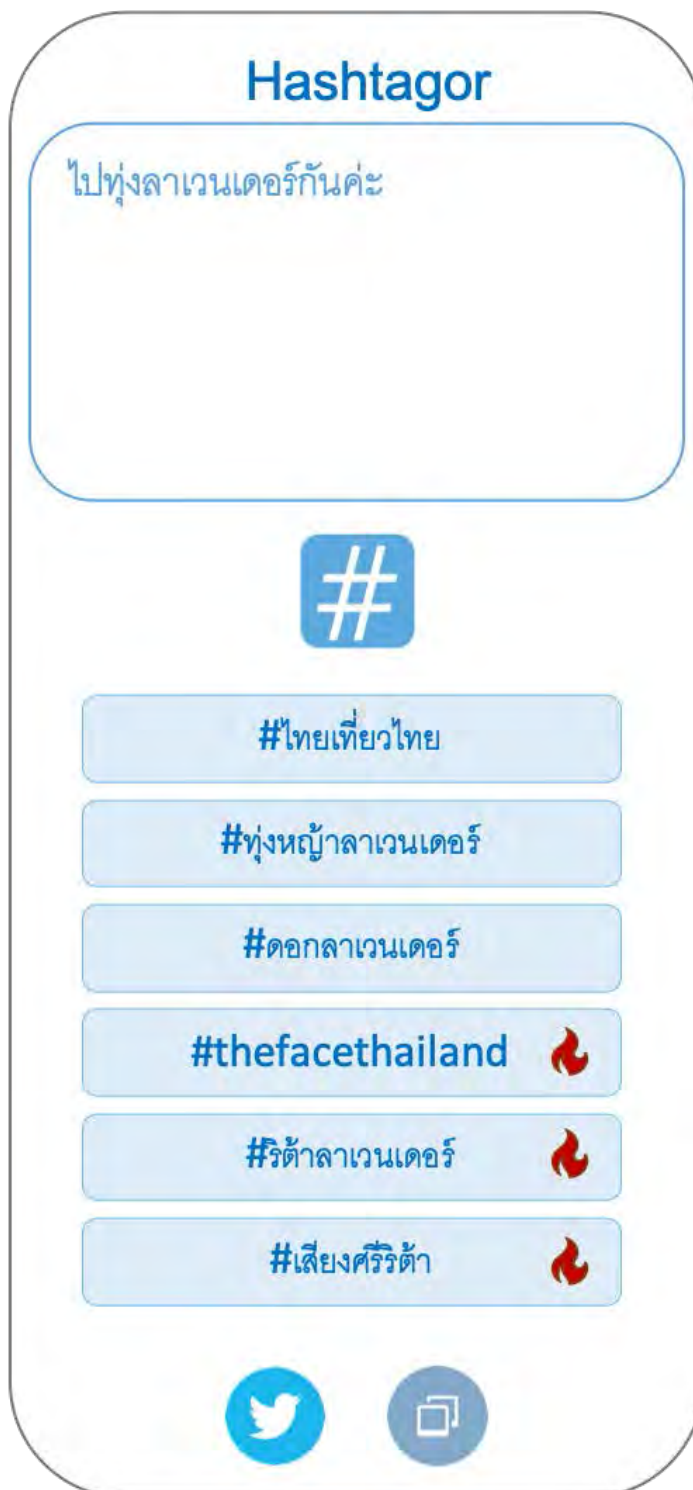


เมื่อใส่ข้อความและกดปุ่มประมวลผล โปรแกรมประยุกต์จะประมวลผล Hashtags ที่สอดคล้องกับข้อความขึ้นมา



### 5.3 หน้าแสดงผลการดำเนินการ

หลังจากกดปุ่มดำเนินการ โปรแกรมประยุกต์จะคำนวณ Hashtag ขึ้นมาทั้งหมด 6 Hashtag โดย 3 ลำดับแรก จะเป็น Hashtag ที่ประมวลผลจากข้อความเท่านั้น และ 3 ลำดับหลังจะเป็น Hashtag ที่เป็นที่นิยมในขณะนั้นและมีความสอดคล้องกับข้อความมากที่สุด





สามารถเลือก Hashtag ที่ต้องการ โดย Hashtag ที่ถูกเลือกจะมีกรอบสีเข้มแสดงว่าเลือกแล้ว จากนั้นสามารถคัดลอกข้อความพร้อมทั้ง Hashtag ที่ถูกเลือกโดยกดปุ่มคัดลอกหรือแชร์ผ่าน Twitter โดยกดปุ่มสัญลักษณ์ Twitter ได้



ผู้ใช้สามารถแก้ไข Tweet ได้ตามปกติ



6. ทดสอบการทำงานของระบบ

7. สรุปผลและเขียนรายงาน

**ตารางเวลาการดำเนินการ**

ขั้นตอนการทำงาน	เดือน/ปีการศึกษา 2561
-----------------	-----------------------

	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. ศึกษาความรู้เกี่ยวกับ Deepcut [3], Tweepy [4], Pandas [5], TensorFlow [6] และ Word2Vec [7]									
2. จัดเตรียมข้อมูล									
3. สังเคราะห์ข้อมูล									
4. ศึกษาความรู้เกี่ยวกับการพัฒนาโปรแกรมประยุกต์บนระบบปฏิบัติการ iOS									
5. พัฒนาโปรแกรมประยุกต์									
6. ทดสอบการทำงานของระบบ									
7. สรุปผลและเขียนรายงาน									

### ประโยชน์ที่คาดว่าจะได้รับ

#### ประโยชน์ที่ได้จากโครงการที่พัฒนาขึ้น

1. ประหยัดเวลาในการคิด Hashtag
2. สามารถใช้เพื่อการตลาดได้
3. สนุกเพลิดเพลินกับโปรแกรม
4. ทำให้ข้อความมียอดเข้าชมเพิ่มขึ้นจาก Hashtag
5. ทำให้ผู้ใช้รู้หัวข้อที่กำลังเป็นที่นิยมใน Twitter

#### ประโยชน์ต่อตัวนิสิตที่ทำโครงการ

1. มีความรู้ความเข้าใจขั้นตอนในการทำงาน ฝึกคิดวิเคราะห์รู้จักการทำงานอย่างมีแบบแผน รู้จักการทำงานเป็นทีม มีวินัย ตรงต่อเวลา และมีความรับผิดชอบต่องาน ฝึกคิดวิเคราะห์ รู้จักทำงานอย่างมีแบบแผน มีระเบียบวินัย ตรงต่อเวลา และมีความรับผิดชอบต่องาน
2. มีประสบการณ์สร้างโปรแกรมบนโทรศัพท์เคลื่อนที่ ทำงานบนระบบปฏิบัติการ iOS รวมไปถึงการใช้ความรู้ด้านการประมวลผลภาษาธรรมชาติ
3. เข้าใจขั้นตอนและสามารถใช้ API ดึงข้อมูลจาก twitter มาได้
4. ได้พัฒนาทักษะการใช้ภาษา Python และได้เรียนรู้วิธีการใช้ไลบรารีต่าง

#### อุปกรณ์และเครื่องมือที่ใช้

## 1. ฮาร์ดแวร์

### 1. MacBook Pro

macOS Mojave 10.14

Processor 2.6 GHz Intel Core i7

Memory 16 GB 2133 MHz LPDDR3

### 2. Google Cloud

Machine type n1-standard-8 (8 vCPUs, 30 GB memory)

GPUs 1 x NVIDIA Tesla P100

### 3. เครื่องพิมพ์

## 2. ซอฟต์แวร์

### 1. Xcode Version 10.1 (10B61)

### 2. ภาษา Python 3.6

### 3. คลังโปรแกรมการเรียนรู้ของเครื่อง ดังต่อไปนี้

1. Deepcut [3]

2. Tweepy [4]

3. Pandas [5]

4. TensorFlow [6]

5. Word2Vec [7]

## 3. วัสดุสำนักงาน

### 1. กระดาษ A4

### 2. หมึกพิมพ์

### 3. หน่วยความจำภายนอก

## งบประมาณ

1. ค่าบริการ Google Cloud Platform

ราคา 2,737 บาท

2. Lightning AV Adapter	ราคา	1,485	บาท
3. Power Adapter Extension Cable	ราคา	738	บาท
4. Seagate Backup Plus Ultra Slim 2.5” 2TB/PL	ราคา	2,650	บาท
5. Type C USB Cable	ราคา	390	บาท
6. ค่าพิมพ์โปสเตอร์และค่าเดินทางไปนำเสนอผลงาน	ราคา	1,000	บาท
7. ค่าถ่ายเอกสารและจัดทำรูปเล่ม	ราคา	1,000	บาท
รวม		10,000	บาท

### เอกสารอ้างอิง

- [1] Simon Kemp. (2018). **DIGITAL IN 2018: WORLD’S INTERNET USERS PASS THE 4 BILLION MARK.** สืบค้นจาก <https://wearesocial.com/blog/2018/01/global-digital-report-2018> [ 14 สิงหาคม 2018 ]
- [2] Dominique Jackson. (2018). **How to Find the Best Twitter Hashtags.** สืบค้นจาก <https://sproutsocial.com/insights/twitter-hashtags/> [ 29 สิงหาคม 2018 ]
- [3] Rakpong Kittinaradorn. (2018). **Deepcut.** สืบค้นจาก <https://github.com/rkcosmos/deepcut> [ 30 สิงหาคม 2018 ]
- [4] Aaron Hill และ Joshua Roesslein. (2018). **Tweepy: Twitter for Python!.** สืบค้นจาก <https://github.com/tweepy/tweepy> [ 1 กันยายน 2018 ]
- [5] pandas-dev. (2018). **pandas: powerful Python data analysis toolkit.** สืบค้นจาก <https://github.com/pandas-dev/pandas> [ 1 กันยายน 2018 ]
- [6] TensorFlow. (2018). **API Document.** สืบค้นจาก [https://www.tensorflow.org/api\\_docs/](https://www.tensorflow.org/api_docs/) [ 1 กันยายน 2018 ]
- [7] Gensim. (2018). **Word2Vec embeddings.** สืบค้นจาก <https://radimrehurek.com/gensim/models/word2vec.html> [ 15 กันยายน 2018 ]

## APPENDIX B

### USER'S MANUAL

#### 1. Start with a splash page

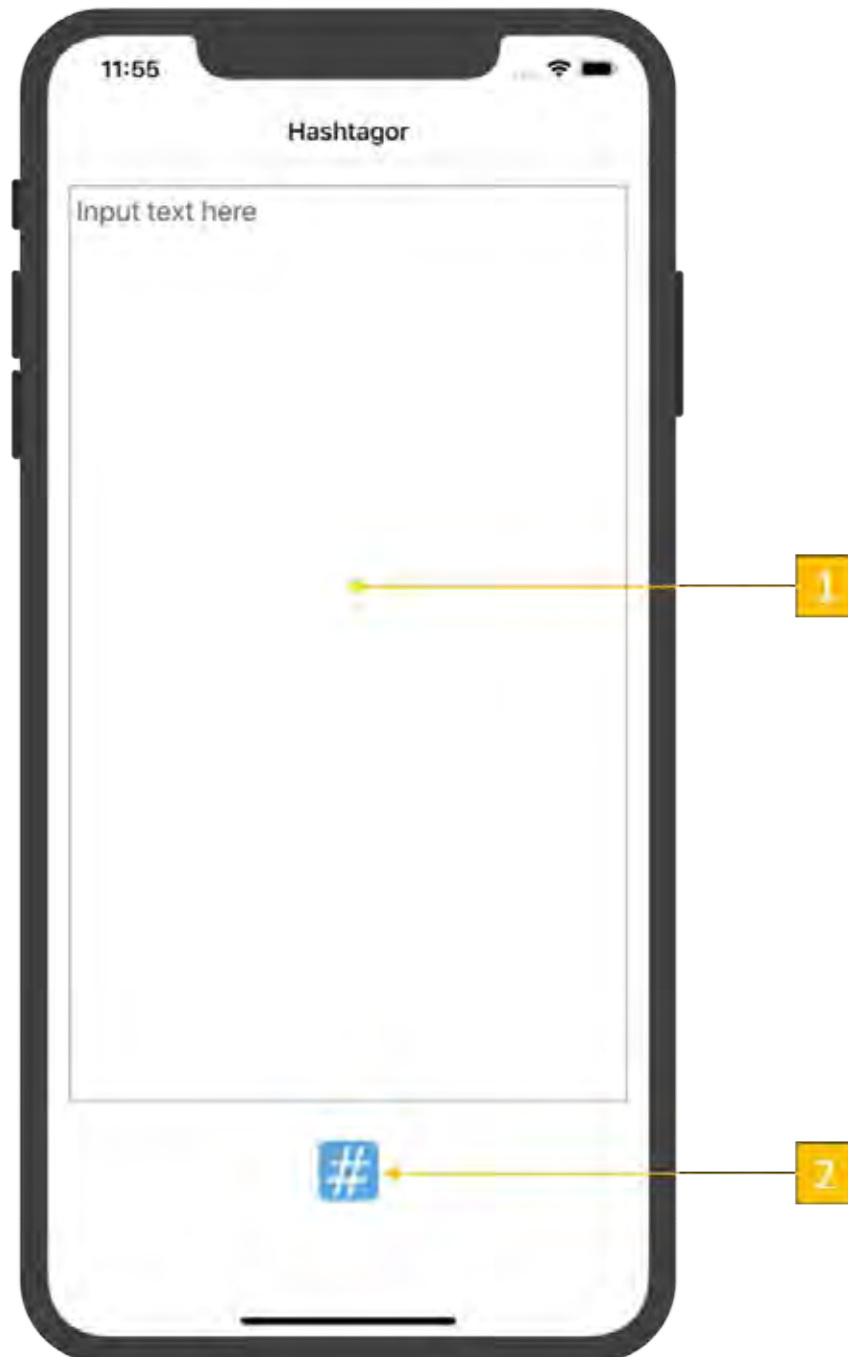
After running Hashtagor then the application will display a splash page (see Figure B.1) while loading, please wait a minute.



**Figure B.1 User's manual: Splash page**

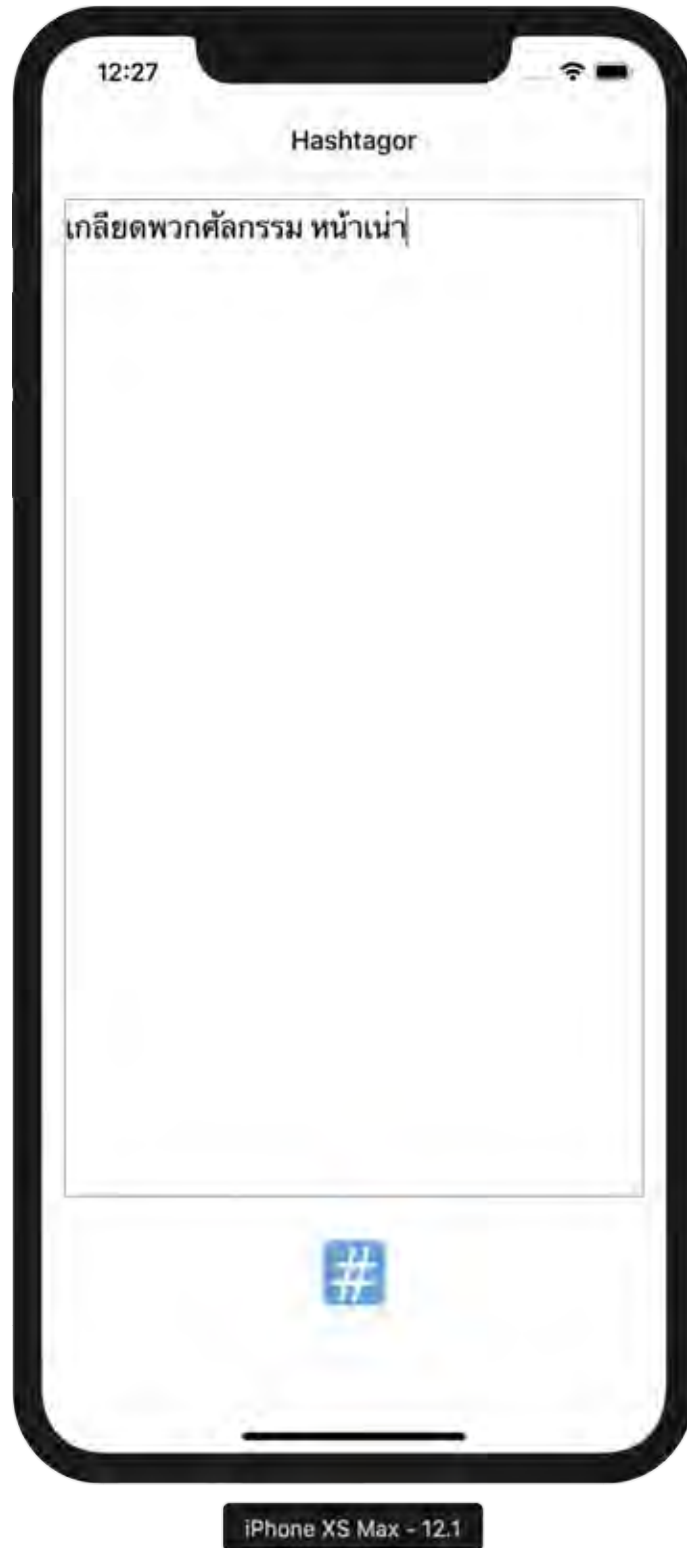
## 2. Welcome to an operation page

In this page (see Figure B.2) a user can input text in the message box (see number 1 in Figure B.2), and when a user finish inputting the text (see Figure B.2), please hit an operation button (see number 2 in Figure B.2).



**Figure B.2 User's manual: Operation page component**

Then the application will be processing the Hashtags that correspond to the message.



**Figure B.3 User's manual: Inserted input example**



If the user does not input any text on the text box but hit an operation button the application will show a message dialog "Please input text" (see Figure B.3).



**Figure B.4 User's manual: Please input text message dialog**

Alternatively, the input text does not have a word in corpus application will show a message dialog "Miss word ..... in the corpus." to notice (see Figure B.5).



**Figure B.5 User's manual: Miss word message dialog**

Alternatively, if the input text is too long, the application will show a message dialog "Text is too long" to notice (see Figure B.6).

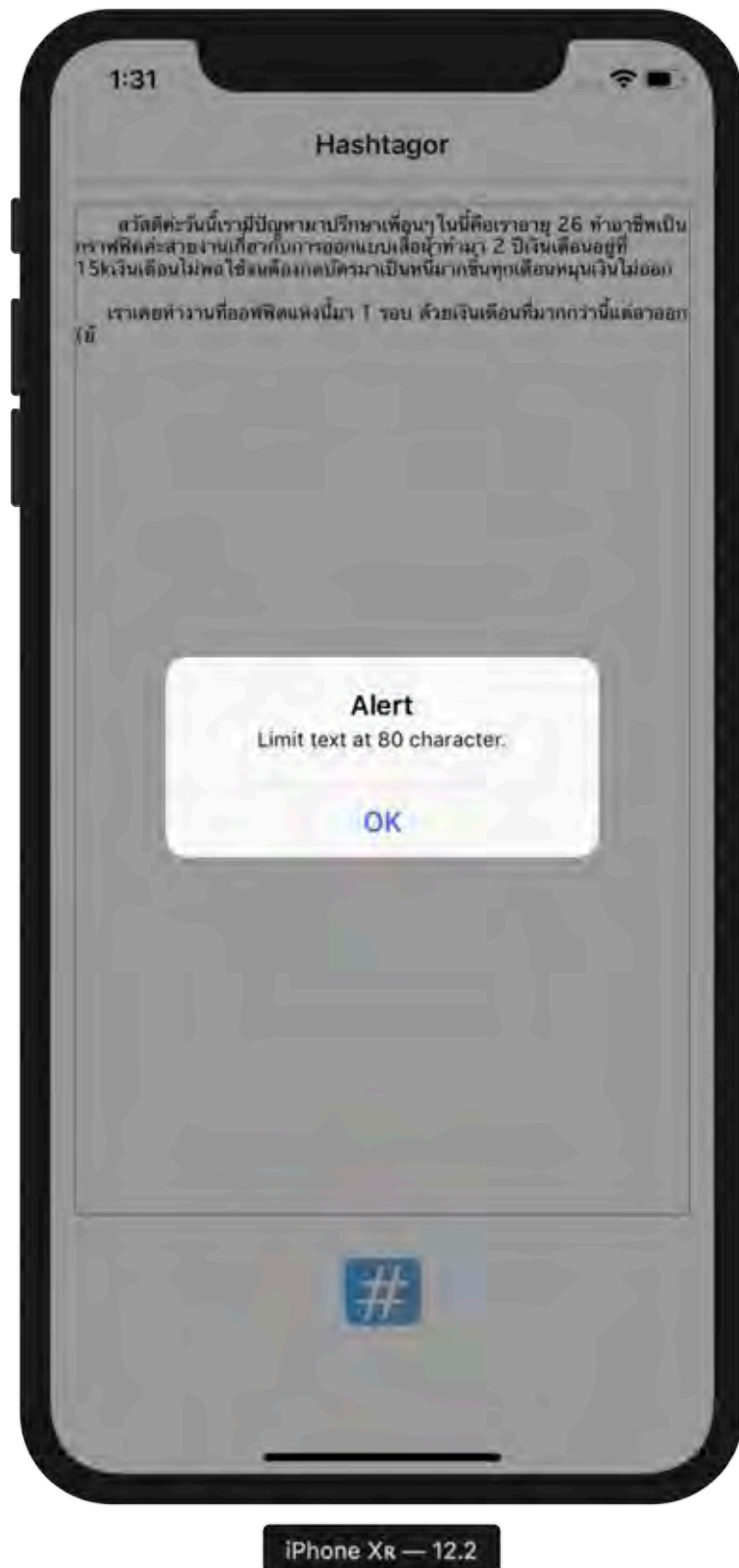


Figure B.6 User's manual: Text too long message dialog example

### 3. Welcome to results page

After pressing the action button, the application will calculate all the 6 Hashtags (see Figure B.7).



**Figure B.7 User's manual: Results page example**

The first three hashtags will be the Hashtag that is processed from the text only and the latter three hashtags with fire icon will be the popular Hashtag at that time and are consistent the input messages

User can select and deselect multiple desired Hashtag. The selected Hashtag will have a check icon showing that selected and show on textbox after text input the same as a standard format on twitter (see Figure B.8).



Figure B.8 User's manual: Selected hashtags example

Users can copy the text and the selected Hashtag to the clipboard by clicking the Copy button. (see Figure B.9).

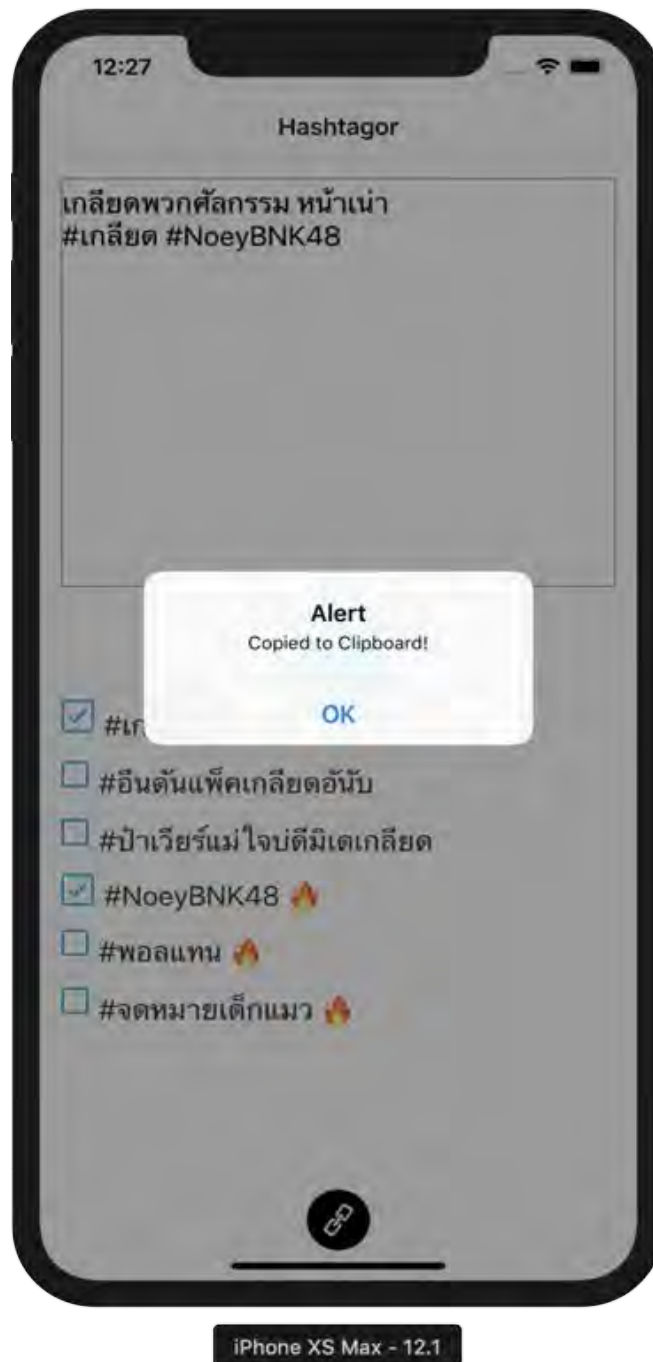


Figure B.9 User's manual: Copy to clipboard success example

Furthermore, User can continue to use the application to discover others hashtag by editing the text in the textbox and hit operation button as usual (see Figure B.10).



Figure B.10 User's manual: Continue using example

## BIOGRAPHY



Chaninchai Hanmuang  
Department of Mathematics and Computer Science  
Faculty of Science, Chulalongkorn University  
Email: [chaninchai.h@gmail.com](mailto:chaninchai.h@gmail.com)



Kwanchanok Srisompong  
Department of Mathematics and Computer Science  
Faculty of Science, Chulalongkorn University  
Email: [Kwanchaaaa@gmail.com](mailto:Kwanchaaaa@gmail.com)