



โครงการ

การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ	ระบบทำนายความนิยมของหัวข้อกระทู้ใน AskReddit และตัวช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติ AskReddit popularity score prediction system with an automatic title refiner
ชื่อนิสิต	นายขจรยศ คำคุณ นายภัทรธร ปัทมสังข์
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2561

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของโครงการทางวิชาการที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของโครงการทางวิชาการที่ส่งผ่านทางคณะที่สังกัด

The abstract and full text of senior projects in Chulalongkorn University Intellectual Repository(CUIR)

are the senior project authors' files submitted through the faculty.

ระบบทำนายความนิยมของหัวข้อกระทู้ใน AskReddit
และตัวช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติ

นายจรรย์ศ คำคุณ
นายภัทรธร ปัทมสังข์

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย


AskReddit popularity score prediction system with an automatic title refiner

Khajornyot Khamkhon
Pattaratorn Pattamangsang

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 ระบบทำนายความนิยมของหัวข้อกระทู้ใน AskReddit และตัวช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติ
โดย นายขจรยศ คำคุณ นายภัทรธร ปัทมสังข์
สาขาวิชา วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงการหลัก อ. ดร.นฤมล ประทานวณิช


ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติ ให้นำโครงการฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงการ วิทยาศาสตร์ (Senior Project)


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

คณะกรรมการสอบโครงการ


.....
(อาจารย์ ดร.นฤมล ประทานวณิช) อาจารย์ที่ปรึกษาโครงการหลัก


.....
(รองศาสตราจารย์ ดร.พีระพนธ์ โสพักสถิตย์) กรรมการ


.....
(ผู้ช่วยศาสตราจารย์ ดร.สมใจ บุญศิริ) กรรมการ

นายขจรยศ คำคุณ, นายภัทรธร ปัทมสังข์: ระบบทำนายความนิยมของหัวข้อกระทู้ใน AskReddit และตัวช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติ. (AskReddit popularity score prediction system with an automatic title refiner) อ.ที่ปรึกษาโครงการหลัก : อ. ดร.นฤมล ประทานวณิช, 48 หน้า.

โครงการเรื่อง “ระบบทำนายความนิยมของหัวข้อกระทู้ใน AskReddit และตัวช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติ” เป็นโครงการที่จัดทำขึ้นเพื่อพัฒนาระบบทำนายความนิยมของหัวข้อกระทู้ใน AskReddit และช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติเพื่อเพิ่มโอกาสที่หัวข้อกระทู้จะได้รับความนิยม โดยการเพิ่มคำ ลบคำ ออก หรือเปลี่ยนคำ ในหัวข้อกระทู้จำนวน 1 คำ ขั้นตอนการพัฒนาประกอบไปด้วย 2 ส่วนหลักๆ คือ ส่วนการพัฒนาแบบจำลองเพื่อทำนายความนิยมหัวข้อกระทู้โดยอาศัยความรู้จากงานวิจัยอื่น และส่วนการพัฒนาตัวช่วยปรับแต่งหัวข้อกระทู้ จากผลของการพัฒนาทำให้ได้แบบจำลองทำนายความนิยมที่มีความแม่นยำอยู่ที่ 71% การวัดผลความถูกต้องของตัวช่วยปรับแต่งหัวข้อกระทู้ทำโดยใช้สถิติจากคณะอักษรศาสตร์เอก ภาษาอังกฤษเป็นคนวัดผล มีอัตราการเพิ่มคำได้ถูกต้องอยู่ที่ 60% อัตราการลบคำได้ถูกต้องอยู่ที่ 71% และ อัตราการเปลี่ยนคำได้ถูกต้องอยู่ที่ 77% ผลลัพธ์จากการปรับแต่งข้างต้นคือรายการประโยคที่ผู้ใช้งานสามารถเลือกไปใช้ได้

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์.....ลายมือชื่อนิสิต ขจรยศ คำคุณ
 ลายมือชื่อนิสิต ภัทรธร ปัทมสังข์
 สาขาวิชา วิทยาการคอมพิวเตอร์.....ลายมือชื่อ อ.ที่ปรึกษาโครงการหลัก นป
 ปีการศึกษา 2561.....

5833611023, 5833650523: MAJOR COMPUTER SCIENCE

KEYWORD: DEEP NEURAL NETWORK / WORD EMBEDDING / BIGRAM / REGRESSION

KHAJORNROT KHAMKHOON, PATTARATORN PATTAMANGSANG:
ASKREDDIT POPULARITY SCORE PREDICTION SYSTEM WITH AN
AUTOMATIC TITLE REFINER. ADVISOR: PROF. NARUEMON
PRATANWANICH, Ph.D., 48 pp.

In this project, we develop a neural network classifier for popularity prediction of the AskReddit post titles. Moreover, we invent an automatic sentence refiner for suggestions of new candidate sentences that are likely to gain more popularity. To perform sentence modification, the refiner makes a single change to the original sentences by word insertion, deletion, and replacement. Any modified sentences with higher predicted probability of becoming popular are suggested as candidate titles. The results show that our model was able to predict the popularity of AskReddit submissions with an accuracy of 71%. We finally carried out an evaluation by human on all suggested sentences. Of all, 77 % of sentences with one word replaced were successfully acceptable, 71% with a word being deleted and 60% with a word being inserted. Those candidate titles can be used as alternatives to their original sentences for AskReddit users.

Department: Mathematics and Computer Science Student's Signature ขจรยศ คำคุณ
Student's Signature ภัทธรณ์ ปัทมมงคล
Field of Study: Computer Science Advisor's Signature น.ป.
Academic Year: 2018

ACKNOWLEDGEMENTS

We would like to express our sincere thanks to our advisor, Dr. Naruemon Pratanwanich and Asst. Prof. Dr. Dittaya Wanvarie for their invaluable help and constant encouragement throughout the planning and development of this project. We are most grateful for their teaching and advice.

In addition, we would also like to thank The National Electronics and Computer Technology Center, National Science and Technology Development Agency, Ministry of Science, Chulalongkorn University for giving us a grant for this project and a good opportunity to participate in the Twenty-first National Software Contest, NECTEC, Thailand: NSC21 in Data Science and Artificial Intelligence Application project”.

Finally, we most gratefully acknowledge our parents and our friends for all their support throughout the period of this project.

CONTENTS

	Page
ABSTRACT IN THAI.....	V
ABSTRACT IN ENGLISH	VI
ACKNOWLEDGEMENTS.....	VII
CONTENTS.....	VIII
LIST OF TABLES	X
LIST OF FIGURES	XI
CHAPTER I INTRODUCTION	1
1.1 BACKGROUND	1
1.2 OBJECTIVES.....	2
1.3 SCOPE.....	2
1.4 PROJECT ACTIVITIES	2
1.5 BENEFITS.....	3
1.6 REPORT OUTLINES	3
CHAPTER II THEORETICAL BACKGROUND.....	4
2.1 DEEP NEURAL NETWORK	4
2.2 NATURAL LANGUAGE PROCESSING.....	4
2.3 WORD EMBEDDING	5
2.4 LANGUAGE MODELS	5
2.5 N-GRAM LANGUAGE MODELS	6
2.6 CROSS-VALIDATION	6
CHAPTER III METHODOLOGY	7
3.1 DATASETS	7
3.2 MODEL ARCHITECTURE.....	9
3.3 MODEL EVALUATION METHOD	12
3.4 PART-OF-SPEECH (POS) TAGGING	12
3.5 THE AUTOMATIC REFINER.....	13
3.5.1 Adding a word to the input sentence	13
3.5.2 Deleting a word in the input sentence	14
3.5.3 Replacing a word with its synonyms	14
3.5.4 Scoring the candidate modified sentences.....	14
CHAPTER IV RESULTS	15
4.1 MODEL LOSS AND ACCURACY	15
4.2 MODIFIED SENTENCES BY THE AUTOMATIC REFINER	18
4.3 STATISTICS OF GENERATED SENTENCES BY THE AUTOMATIC REFINER.....	20
4.3.1 Single-word insertion	20
4.3.2 Single-word deletion.....	21
4.3.3 Single-word replacement.....	22
4.4 HUMAN EVALUATION.....	22

CONTENTS (CONT.)

	Page
CHAPTER V DISCUSSION.....	24
REFERENCES.....	26
APPENDIX A THE PROJECT PROPOSAL OF COURSE 2301399 PROJECT PROPOSAL.....	28
BIOGRAPHY.....	36

LIST OF TABLES

	Page
Table 1.1 Schedule table of project activities.....	2
Table 3.1 Details of the first 4 records with the highest scores in the dataset.....	7
Table 3.2 Statistics of the dataset.....	8
Table 3.3 Model Hyperparameters used in the model training procedure.....	12
Table 3.4 Results of part-of-speech tagging of the example sentence.....	13

LIST OF FIGURES

	Page
Figure 3.1 Model architecture for embedding AskReddit titles	9
Figure 3.2 Heatmap of the relationship between an hour in day and day of week	10
Figure 3.3 Model architecture for embedding submission date properties.....	10
Figure 3.4 The final model.....	11
Figure 4.1 Training and testing loss of the model without the timing features.....	16
Figure 4.2 Training and testing loss of the model with the timing features.....	16
Figure 4.3 Training and testing accuracy of the model without the timing features.....	17
Figure 4.4 Training and testing accuracy of the model with the timing features.....	17
Figure 4.5 Outputs from adding a word to the input sentence.....	18
Figure 4.6 Outputs from removing a word from the input sentence.....	19
Figure 4.7 Outputs from replacing the word “good” in the input sentence.....	19
Figure 4.8 The distribution over the number of candidates generated per an input sentence in case of a word being added	20
Figure 4.9 The distribution over the number of candidates generated per an input sentence in case of a word being removed	21
Figure 4.10 The distribution over the number of candidates generated per an input sentence in case of a word being substituted.....	22
Figure 4.11 The number of successful and unsuccessful candidate sets, categorized by the edition methods.....	23

CHAPTER I

INTRODUCTION

This chapter gives the background knowledge, objectives, scope, project activities, benefits and report outlines of our project.

1.1 Background

Forum is an internet site where users can post topics for discussion and exchange their opinions. While Pantip is one of the most popular forums for Thai, Reddit attracts more users internationally.

Reddit is ranked at the 18th place for the most popular websites in the world with the highest number of active users. Each day, there are over 1 million new posts on the forum, and the number of replies or comments exceed 5 million. In 2018, Reddit has approximately 330 million users [1]. This is very massive. Reddit organizes its posts based on user-created areas of interest into categories called “subreddit”. Examples of subreddits are “fitness”, “politics”, “AskReddit”, etc. Moreover, each post in a subreddit has a score which indicates its popularity. Users can up-vote or down-vote a post to increase or decrease its popularity score by 1 respectively. High-scored posts are shown on the front page of subreddit for a certain amount of time. This leads to an increasing number of comments in the posts.

The score of each post varies depending on many factors, such as the title of a post, the content of a post (which can be plain text or image) or even a post’s submission time. All of these factors are difficult to analyze together. For simplicity, we choose the subreddit called “AskReddit” which is the 3rd most popular among all subreddits [2]. It is one of the few subreddits where users are allowed to post only the questions they want to ask as titles. This type of post has no content so we can imply that the score for each post is influenced by the sentence used in title alone with some biases from its submission time, which we will take into account in this project.

Leaving aside the post’s submission time, we assume that a sentence used as a post title has some properties that attract people to and give a positive vote for the post, such as grammar, context and the use of words. Fortunately, sarcasm sentences are not allowed in the AskReddit so we can analyze data in a direct approach.

In this project, we aim to develop the AskReddit popularity score prediction system, as well as an automatic title refiner. The system receives an English sentence as an input. It estimates the chance a post is likely to be popular on AskReddit if it is posted as a title. After that, the results, which contained submission title and the predicted score, are passed to an automatic title refiner. This refiner performs three the following methods to edit the sentence; adding a word, deleting a word and replacing a word with a new word. The final outputs yield a candidate sentence that, when used as a title, has more chance to make a post become popular relative to the user-input sentence based on our prediction system. We apply machine learning and deep learning techniques to achieve these prediction tasks.

1.2 Objectives

1. To develop the AskReddit popularity score prediction system.
2. To develop an automatic title refiner that can suggest one or more candidate sentences that are likely to gain higher popularity score, if any, by i) adding a word, ii) deleting a word and iii) replacing a word with a new word to a given input sentence.

1.3 Scope

1. We only consider subreddits with no content in the posts. In this project, we use a subreddit called “AskReddit”.
2. The title must be in English, and the automatic refiner only considers the English sentence as valid input.
3. The automatic title refiner is to make a single change to the input sentence at a time, which is adding a word, deleting a word or replacing a word with a new word. If the refiner does not find any candidate sentence that the estimated score is higher than the input sentence’s, no change will be suggested.

1.4 Project Activities

1. Study previous research regarding Reddit popularity score prediction.
2. Collect and prepare the related data, such as Reddit submission data from AskReddit.
3. Design and implement a Deep neural network model
 - 3.1. Develop the model
 - 3.2. Evaluate the model
 - 3.3. Improve the model
4. Conclude the results and write up.

Our detailed plan based on the project activities described above is shown below.

Table 2.1 Schedule table of project activities

Procedure	2018					2019			
	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
1.Study previous research in Reddit popularity score prediction									
2.Obtain data									
3.Analyze and design the Deep learning model									
3.1.Develop the model									
3.2.Measure the performance of the model									
3.3.Improve the model									
4.Conclude the results and prepare documentation									

1.5 Benefits

Benefits for the students who implement this project.

1. Get to learn more about Machine learning and Deep learning
2. Get to learn more about related mathematics (e.g. Statistic)
3. Understand how to work as a team
4. Gain experience in programming from working with Python and Cloud Computing

Benefits of the project

1. Present the automatic refiner system by using knowledge from the previous research works.
2. Users can learn knowledge about the properties that affect the post's popularity in AskReddit.
3. Users can use the system developed from this project to refine their sentence before submitting a post on AskReddit, leading to a better chance that the submitted post will be popular.
4. The developed system can be used in marketing.

1.6 Report Outlines

This report consists of five chapters as follows. Chapter I includes background, objectives, scope, project activities and benefits of this project are presented. Chapter II includes the theoretical background knowledge relating to the work in this project are reviewed. Chapter III includes we present the methodology including how we create the model and automatic refiner system. Chapter IV includes the model's performance is evaluated and results from automatic refiner system are reviewed. Chapter V includes the results of our project, as well as problem, obstacles, and future works, are discussed and concluded.

CHAPTER II

THEORETICAL BACKGROUND

This chapter provides the introduction of deep neural networks and discusses the concept of natural language processing, word embedding techniques, language models and N-gram models.

2.1 Deep Neural Network

A deep neural network (DNN) is an artificial neural network with multiple layers between the input and output layers (Sze, Vivienne, et al., 2017). A DNN turns inputs into outputs by estimating the mathematical mapping function in the form of weights between neurons, with defined forms of activation functions.

A DNN can be used in many different applications, for example

- Recommendation engines: DNNs were applied for classification and regression tasks to provide a better user experience and service.
- Text sentiment analysis: recurrent neural networks, sequence-based DNNs, are used to extract high-level information, which is further used for evaluating the meaning at the sentence level.
- Chatbot: designed to simulate conversation with human users over the internet.
- Image recognition: uses convolution neural networks to classify images, which enables many further downstream applications: sorting image, finding similarities between images, and recognizing human faces.

In this project, we will focus on the text sentiment analysis, and how we use DNNs and natural language processing to predict the popularity of posts in AskReddit.

2.2 Natural Language Processing

Natural language processing (NLP) is a subfield of Artificial Intelligent (AI). It enables computers to understand human language.

Since human language is more complex and abstract, the learning process in NLP can be categorized into six levels as follows.

1. Morphological Level: to understand alphabets and be able to distinguish differences between consonants and vowels.
2. Lexical Level: to understand words and their meaning.
3. Syntactic Level: to understand sentences and how to construct new sentences.
4. Semantic Level: to understand contexts of sentences; how words change their meanings depending on the structure of sentences they sit in.
5. Discourse Level: to understand the relation between sentences, how sentences are connected.

6. Pragmatic Level: to understand words and sentences by referring to past knowledge, which is sometimes not explicitly present in the current contents. In this level, NLP is able to interpret the meaning of sentences close to human's level.

2.3 Word Embedding

Word embedding is a process to map words or phrases to numeric vectors. It uses a mathematical embedding from a binary vector with the vocabulary size to a continuous vector space with lower dimension (Mikolov, Tomas, et al., 2013).

A basic approach to word embedding is known as one-hot encoding, where word is to be represented by a binary vector. The length of the one-hot vector is equal to the size of the vocabulary. For example, the sentence "I am a student." can be represented by a 4-dimensional vector where 'I' equals to [1,0,0,0], 'am' equals to [0,1,0,0], 'a' equals to [0,0,1,0] and 'student' equals to [0,0,0,1]. From this example, it is clear that one-hot encoding only represents the existence of words in a document, but it does not capture their meanings or contexts in the sentences as the distance between words in this embedded space are equal.

2.4 Language models

Language models are models that tell probabilities of a sequence of words, or probabilities of an upcoming sequence over words based on the information of previous words. There are two types of language models: N-gram language model, and grammar-based language model such as probabilistic context-free grammar (PCFGs) (Ajitesh, Kumar, 2018).

There are two main approaches to create a language model.

The first method is to calculate the probability of a sequence of words based on the product of a probability of each word.

The probability of occurrence of a sentence consisting of n words can be calculated in equation (1).

$$P(\text{Sentence}) = P(w_1) * P(w_2) * P(w_3) * \dots * P(w_n) \quad (1)$$

where the probability of each word $P(w_1)$ can be calculated as in the equation (2).

$$P(w_i) = \frac{c(w_i)}{c(w)} \quad (2)$$

where w_i is a word i , $c(w_i)$ is the number of times w_i occurs in the corpus and $c(w)$ is the number of words in the corpus.

The other method is to calculate the probability of a sequence over words based on the product of probabilities of words conditioned on previous words.

The probability of occurrence of a sentence containing n words computed as in equation (3)

$$P(\text{Sentence}) = P(w_1 | [\text{StartOfSentence}]) \quad (3)$$

$$P(w_2 | w_1)P(w_3 | w_2) \dots P(w_n | [\text{EndOfSentence}])$$

where the probability of a word given its precedent word can be calculated in Equation (4).

$$P(w_i | w_{i-1}) = \frac{P(w_{i-1} w_i)}{P(w_{i-1})} \quad (4)$$

2.5 N-gram language models

N-gram are language models that determine probability based on the count of the sequence of words. The model can be unigram, where only a word of interest alone is taken into account, Bigram, where only two words of interest alone are taken into account, Trigram, where only three words of interest alone are taken into account, and so on.

Let w_i denote the i^{th} word in a sentence. Consider we have a sentence below.

$$Sentence = w_1 w_2 w_3 \dots w_n.$$

Based on a bigram model, the probability of this sentence can be calculated as in Equation (5).

$$P(Sentence) = P(w_1|[StartOfSetence])P(w_2|w_1)P(w_3|w_2) \dots P([EndOfSentence]|w_n)$$

where

$$P(w_2|w_1) = \frac{P(w_1 w_2)}{P(w_1)} = \frac{C(w_1 w_2)}{C(w_1)} \quad (5)$$

That is, the probability of a word “ w_2 ” given a word “ w_1 ” has occurred is the probability of “ w_1 ” and “ w_2 ” occurred together divided by the probability of occurrence of “ w_1 ”, which is equivalent to the count of “ $w_1 w_2$ ” divided by the count of “ w_1 ”.

This is categorized into the second method of language models that we have discussed in the previous section. Thus, the probability of a word w_i given the previous word w_{i-1} can be calculated as the following.

$$P(w_i | w_{i-1}) = \frac{P(w_{i-1} w_i)}{P(w_{i-1})} .$$

2.6 Cross-validation

In Machine learning, we usually split the data into three subsets; training, validation and test sets. Generally speaking, training data are used to fit the model parameters. Validation data are used for hyperparameter tuning where a different combination of hyperparameter values are evaluated until the best-trained model is found. Test data are used to evaluate the performance of the final model which is used to compare with different methods in an unbiased way.

CHAPTER III METHODOLOGY

This chapter shows the details of datasets used, the deep learning model, our experiment setup, and our automatic refiner development.

3.1 Datasets

We used the dataset from subreddit AskReddit (<https://www.reddit.com/r/AskReddit>). We used Google BigQuery, a Python library, to query all the posts dated between 1st September 2018 and 31st December 2018. The dataset we obtained contains 503,938 records, each of which has 7 columns, as shown in Table 3.1.

Table 3.1 Details of the first 4 records with the highest scores in the dataset.

id	title	hour	minute	dayof week	dayof year	score
9gx68i	How would you feel about a law that requires people over the age of 70 to pass a specialized driving test in order to continue driving?	14	1	2	261	149070
9hef7a	In a video game, if you come across an empty room with a health pack, extra ammo, and a save point, you know some serious shit is about to go down. What is the real-life equivalent of this?	7	17	4	263	83296
9icx7a	What is a website that everyone should know about?	19	16	0	266	82665
9jiras	What could the U.S.A. have spent \$1,000,000,000,000 on instead of a 17 year-long war in Afghanistan?	6	17	5	271	74998

Each row has a unique id consisting of 6 characters, a question title, submission date broke down into 4 properties: hour, minute, day of week and day of the year. Hour and minute

indicate the time of the day starting from 00:00 AM. Day of week is 0 for Sunday, 1 for Monday, and so on. Day of the year is a number of days passed in a year ranging from 1st January.

We classified the posts into two categories: popular and unpopular based on the percentile of their popularity score. Posts scored higher than 50% of the rest were classified as popular, and those scored less than 50% of the rest were classified as unpopular. From the 503,938 records, there were 208,860 records classified as popular and 295,078 records classified as unpopular. This caused an imbalanced dataset, which is a major problem in a classification task. To address this problem, we balanced the popular and unpopular ratio by removing 86,218 records from unpopular records. The final dataset has 417,720 records. The data statistics are shown in Table 3.2.

Table 3.2 Statistics of the dataset.

	score
mean	25.793
std	815.115
25%	1.000
50%	1.000
75%	3.000
min	0
max	149070

From table 3.2, the mean of the score is 25.793. The standard deviation of the score is 815.115. The 25th and 50th percentile of the score are 1. The 75th percentile of the score is 3. The minimum score is 0 and the maximum score is 149,070.

3.2 Model Architecture

We used the model from Max Woolf, a Data Scientist at BuzzFeed in San Francisco [cite]. This model takes each AskReddit titles up to 20 words as an input. Each word is then mapped to a 50-dimensional vector in the embedding layer, initialized using a look-up table of 40,000 words from the pre-trained GloVe word embedding model (see Chapter 2.x). All of these embedded word vectors are averaged together in global average pooling layer to reduce variance and computational complexity. Figure 3.1 illustrates the network architecture for embedding the AskReddit titles.

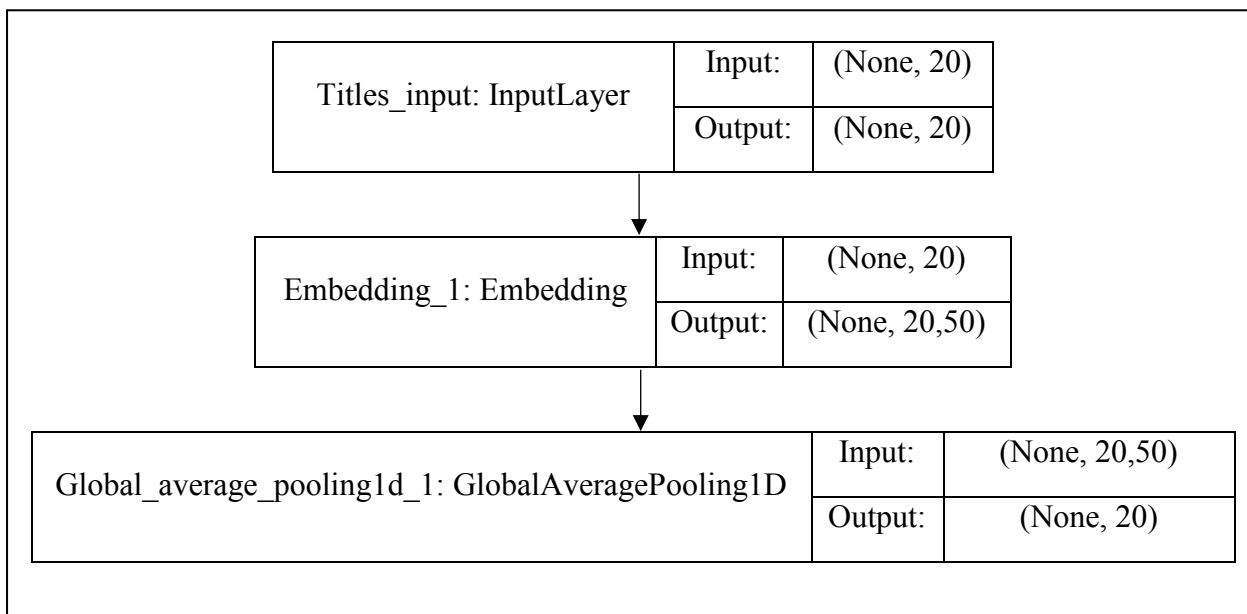


Figure 3.1 Model architecture for embedding AskReddit titles.

Since we found that the submission date had an influence on the popularity of AskReddit posts, we incorporated the submission date into the score prediction model. Although the popularity scores are not clearly different among days in a week, they are varied from hours to hours of a day, in particular, from 6 AM to 9 AM (see Figure 3.2). According to Woolf's model, each date property (hour, minute, day of week, day of year) has its own embedding layer, which outputs a 64-dimensional vector as shown in Figure 3.3. This helps the model learn hidden characteristics behind the submission date from a traditional one-hot encoding method.

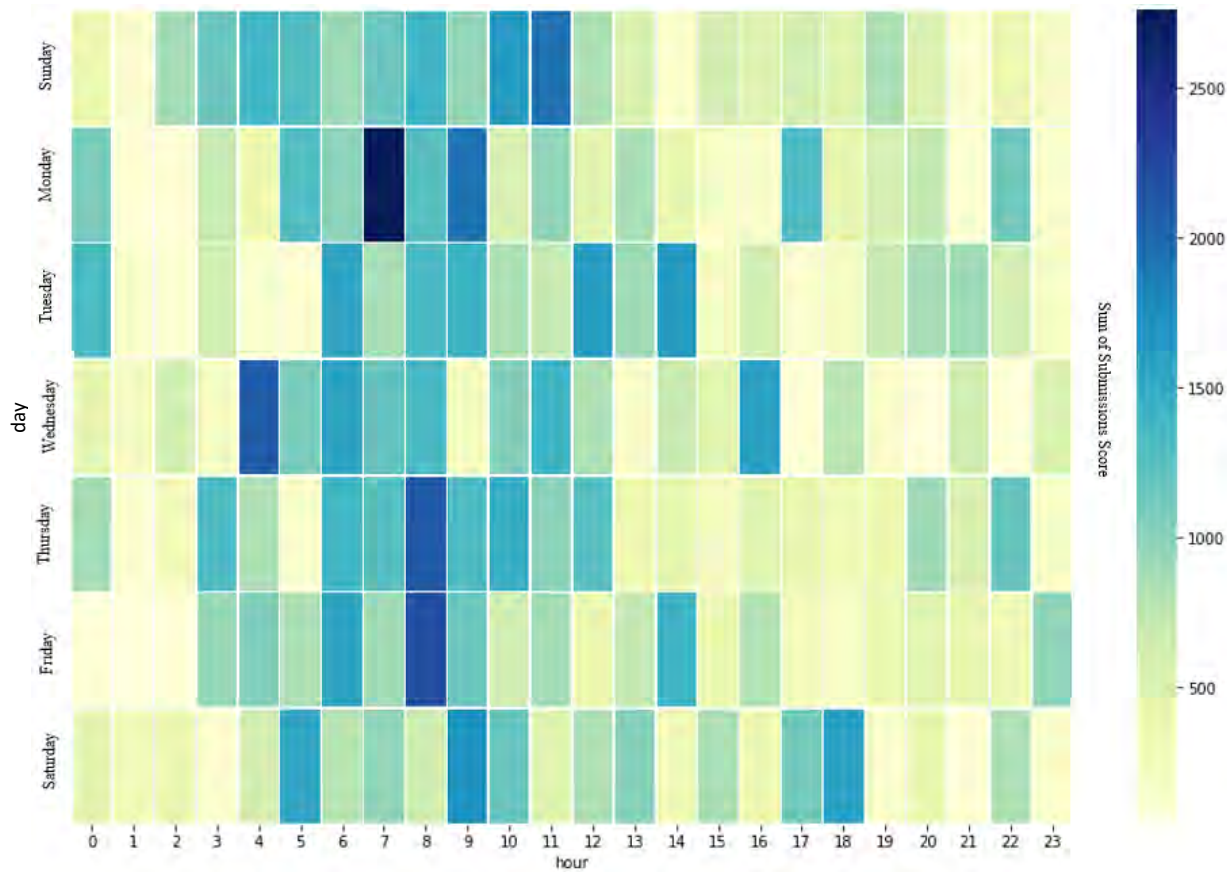


Figure 3.2 Heatmap of the relationship between an hour in day and day of week.

Each entry is the sum of popularity scores of all posts submitted in a particular hour and day of week. A higher value is denoted by a darker color.

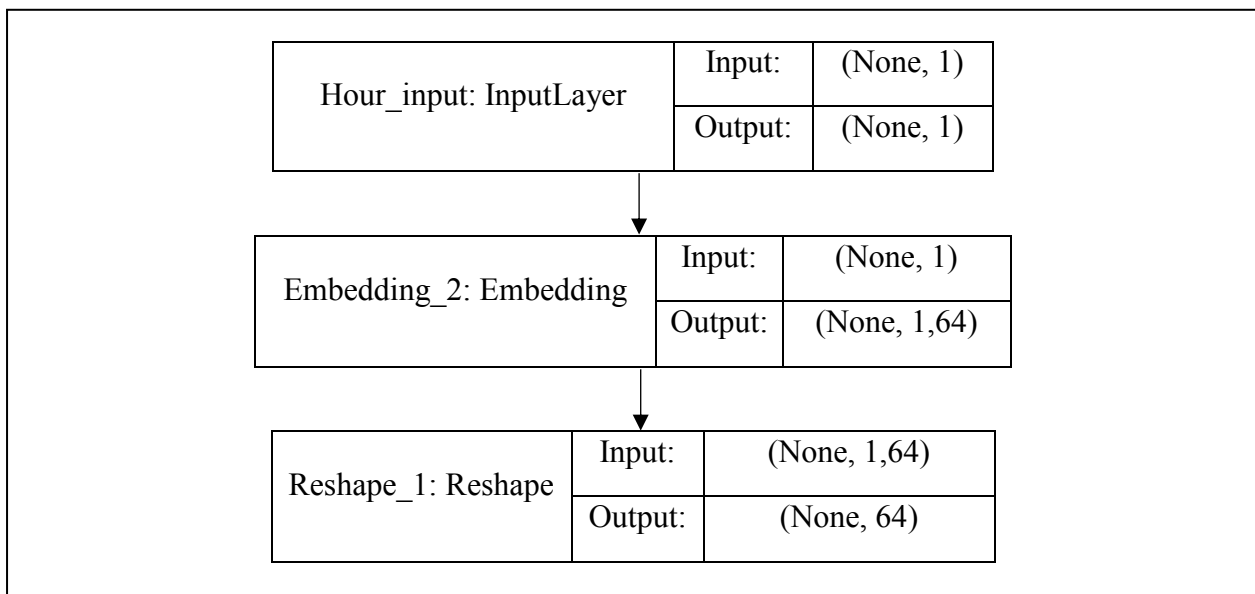


Figure 3.3 Model architecture for embedding submission date properties.

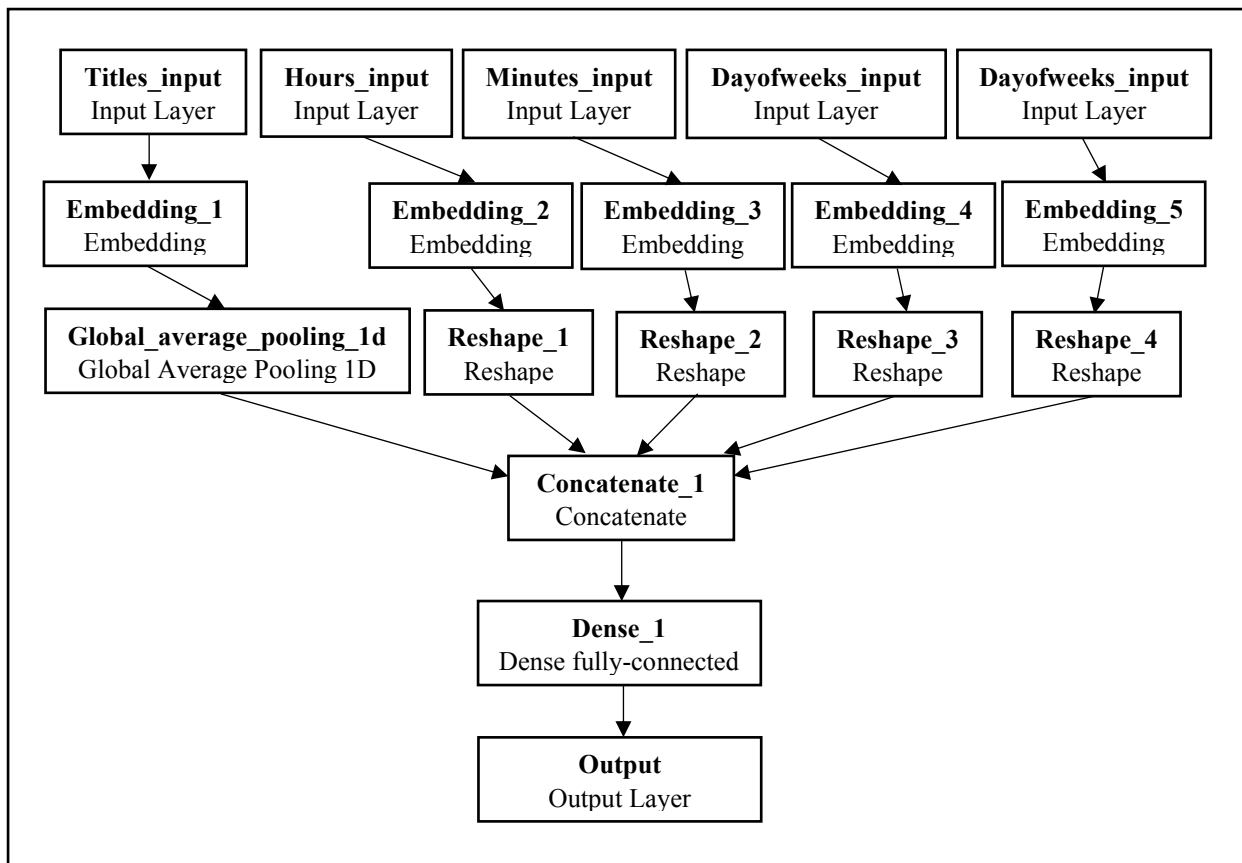


Figure 3.4 The final model

After embedding layers, the 50-dimensional vector from the global average pooling layer is concatenated to the four 64-dimension vectors from each of timing feature, resulting in a 306-dimensional vector. This vector is connected to a dense fully-connected layer to find hidden interactions among all 5 input features (title, hour, minute, day of week and day of the year). Finally, the model outputs a probability of how likely the given submission title is popular.

We decided the number of epochs to train the model using an early stopping method. Training will stop when the loss stops decreasing. Moreover, we tested The other hyperparameters were selected by testing the parameters with different values and choosing the best outcome. The list of hyperparameters used in this work is shown in Table 3.3.

Table 3.3 Model Hyperparameters used in the model training procedure.

Hyperparameters	Parameter Value
Number of Cells in Input Layer	20
Word Embedding Dimension	50
Timing Feature Embedding Dimension	64
Number of Cells in Dense-fully Connected Layer	256
Number of Epochs	5
Dropout Rate	0.2

3.3 Model evaluation method

We used a train-test split method with a 4:1 ratio instead of k-fold cross-validation. The reason is that the size of datasets (417,720 records) is too large. It would take much more time to process with k-fold cross-validation. The train-test split method is sufficient and still able to test the popularity score prediction.

3.4 Part-of-speech (POS) tagging

POS tagging is required for our automatic refiner system to be able to process text data. Without POS tagging, the refiner system will not understand the context of words in a sentence nor enable to suggest changes properly. We performed a part-of-speech tagging (POS) to the input sentences using the natural language toolkit (NLTK) library. For example, the sentence “What is the smallest thing that makes you lose your temper immediately?” yielded the following results shown in Table 3.4.

**Table 3.4 Results of part-of-speech tagging of the example sentence
“What is the smallest thing that makes you lose your temper immediately?”**

Word	POS Tagging	Meaning
what	WP	wh-pronoun (who, what)
is	VBZ	the present form of the verb
the	DT	determiner
smallest	JJS	superlative adjective
thing	NN	singular noun
that	WDT	determiner
makes	VBZ	the present form of the verb
you	PRP	personal pronoun
lose	VB	the base form of the verb
your	PRP\$	possessive pronoun
temper	NN	singular noun
immediately	RB	adverb

3.5 The automatic refiner

The refiner takes the original sentence with its POS tags as an input and performs three of the following methods to modify the original sentence; adding a word, deleting a word and replacing a word with its synonyms. All the modified sentences are finally re-scored by the prediction model explained in Section 3.3 and those with the scores higher than the original sentences are suggested as alternative candidates to users.

3.5.1 Adding a word to the input sentence

We used a bi-gram model for word addition. To construct the bi-gram model, we used datasets from Brown University Standard Corpus of Present-Day American English (Brown corpus), which were compiled from works published in the United States in 1961, containing approximately one million words (Kirsten Malmkjær, 2013). Our refiner begins with iterating through each position in the input sentence to add a new word. The first position of the input sentence where a question word is located is always skipped. Given the word of each position I , the refiner looks for the word listed in the bigrams with the highest possibilities of occurrence. In this process, we used POS tags to ensure that the added word has a correct part of speech. For example, if the given word is “thing” which is a noun, the word before a noun should be adjective. The refiner ignores words in the bigrams that are not adjective, accordingly.

3.5.2 Deleting a word in the input sentence

To delete a word, the refiner iterates through each word in the input sentence to remove if it is either an adjective in front of a noun or adverb since these types of words can be removed with little violation of the English grammar.

3.5.3 Replacing a word with its synonyms

We used the concept of word synonyms to implement this type of modification. Synonyms are words or phrases that have exactly or nearly the same meanings. We used a synonym database from the Natural Language Toolkit (NLTK). The refiner first iterates through each word in the input sentence and then replace it with synonyms if it is a noun, adjective, adverb or verb that is not an auxiliary or modal verb.

3.5.4 Scoring the candidate modified sentences

All of the sentences that are successfully modified are then fed into the learned prediction model for score estimation. The candidate sentences with the predicted probability higher than that of the input sentence are finally suggested to users.

CHAPTER IV RESULTS

This chapter shows the results from our trained model, outputs from the automatic refiner with their, and results from the human evaluation.

4.1 Model loss and accuracy

We compared the models which were trained on two different input sets i.e. titles with timing features and without timing features. Both training processes were stopped after five epochs as a result of early stopping. We compared the loss and accuracy of both models.

Figure 4.1 shows the loss values of the model without the timing features. For the training set, the loss values decrease from 0.6165 to 0.5191 as the number of epoch goes up from 1 to 5. However, for the test set, the loss values decrease from 0.5932 to 0.5921 for the first two epochs but increase to 0.5957 at the fifth epoch. This indicates an overfitting problem. In other words, the model seemed to remember the training data rather than learning from them since the third epoch.

Figure 4.2 shows the loss values of the model with the timing features. For the training set, the loss values decrease from 0.5921 to 0.5320 over epochs. For the test set, the loss values also decrease from 0.5604 to 0.5536 throughout the running epochs. These results show no sign of overfitting.

Figure 4.3 shows that the model with the timing features achieves the accuracy of 71.40% on the training set and the 69.72% on the test set. The accuracy in the test set is lower than the training set, which indicates that there is an overfitting problem but not by a significant amount.

Figure 4.4 shows that the model without the timing features achieves accuracy of 70.33% in training sets and 68.42% in the test set. The accuracy in the test set is lower than the training set, which indicates that there is an overfitting problem but not by a significant amount.

Comparing the results from figure 4.3 to 4.4, the model with the timing features has higher accuracy than the model without the timing features by 1.07% for the training set and 1.30% for the test set.

It can be concluded that the model with the timing features performed better than the model without the timing feature. This confirms our observation that the submission date information has an impact on the popularity of the post. Therefore, we used the model with the post's timing properties for later procedures in this work.

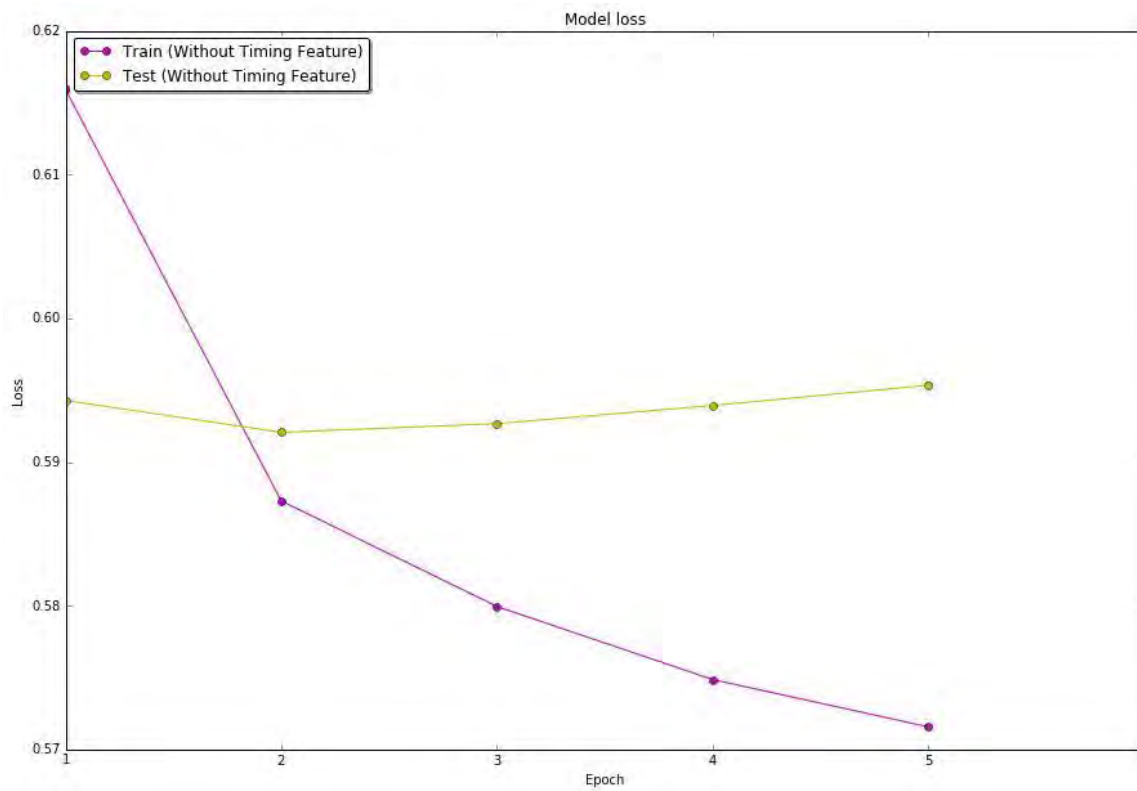


Figure 4.1 Training and testing loss of the model without the timing features

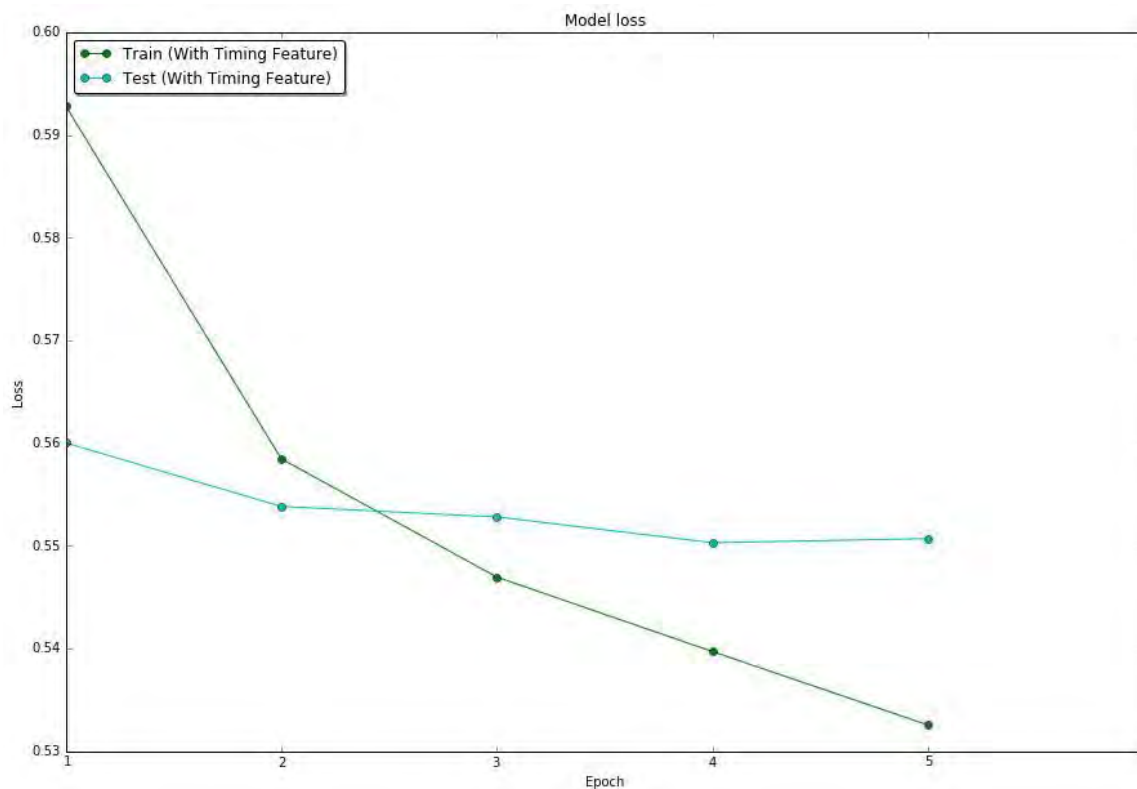


Figure 4.2 Training and testing loss of the model with the timing features

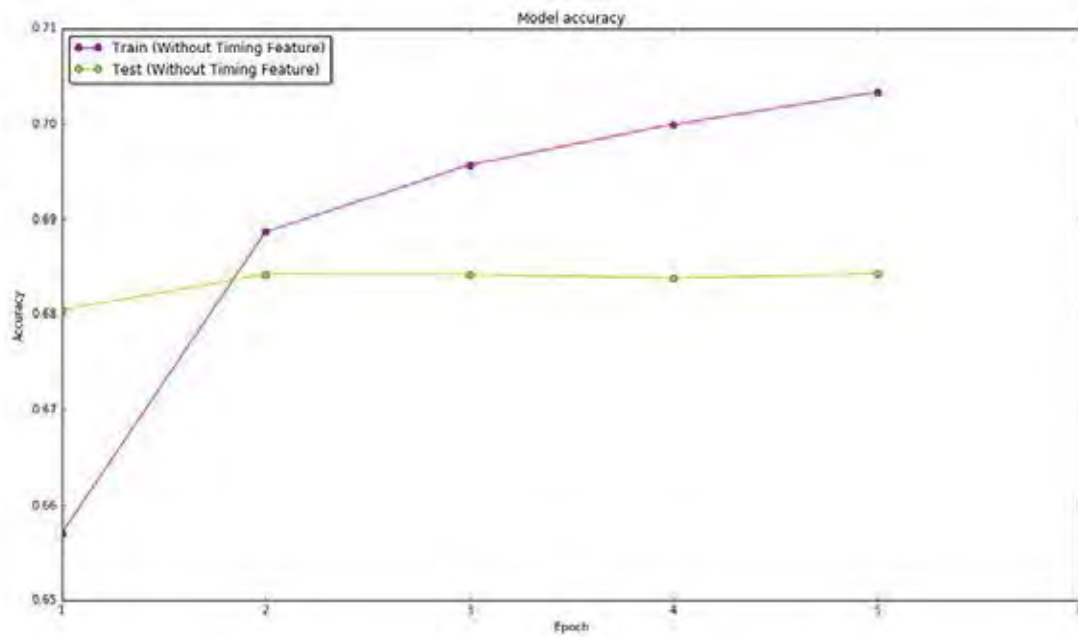


Figure 4.3 Training and testing accuracy of the model without the timing features

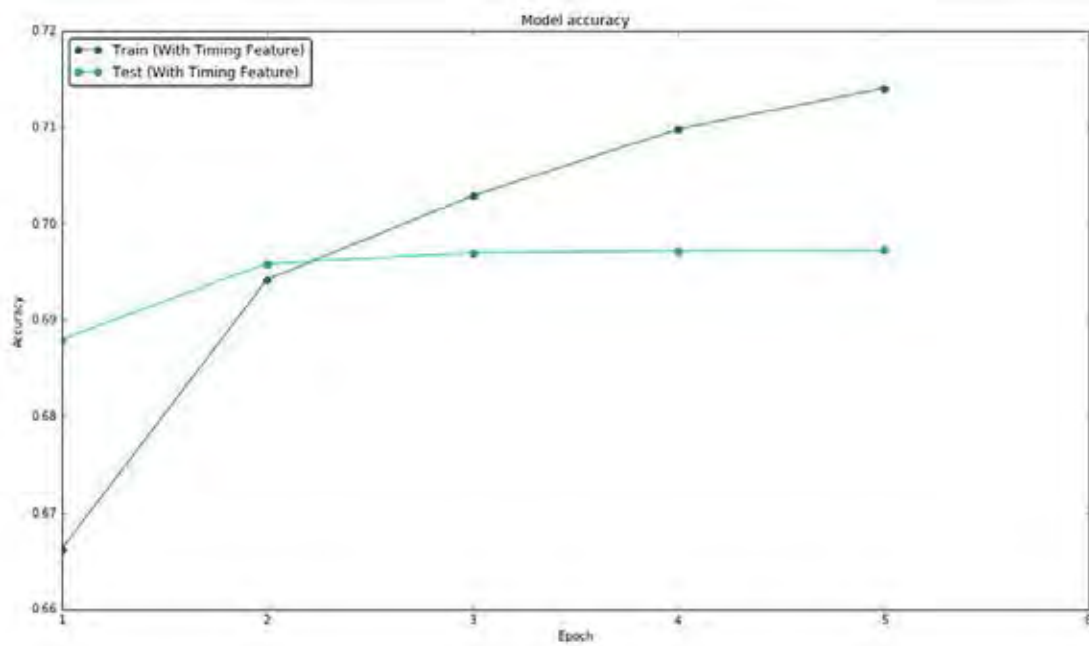


Figure 4.4 Training and testing accuracy of the model with the timing features

4.2 Modified sentences by the automatic refiner

The system outputs new sentences after executing the three types of modification: adding a word, deleting a word and replacing a word with its synonyms, if those sentences have a score higher than the input sentences. Note that there can be none or multiple candidate sentences resulted from each modification.

For example, if the input sentence is “What is a good thing someone can do to better their life?” which has the estimated probability of 0.691. By editing this sentence, the automatic refiner system suggests the outputs shown in Figure 4.5.

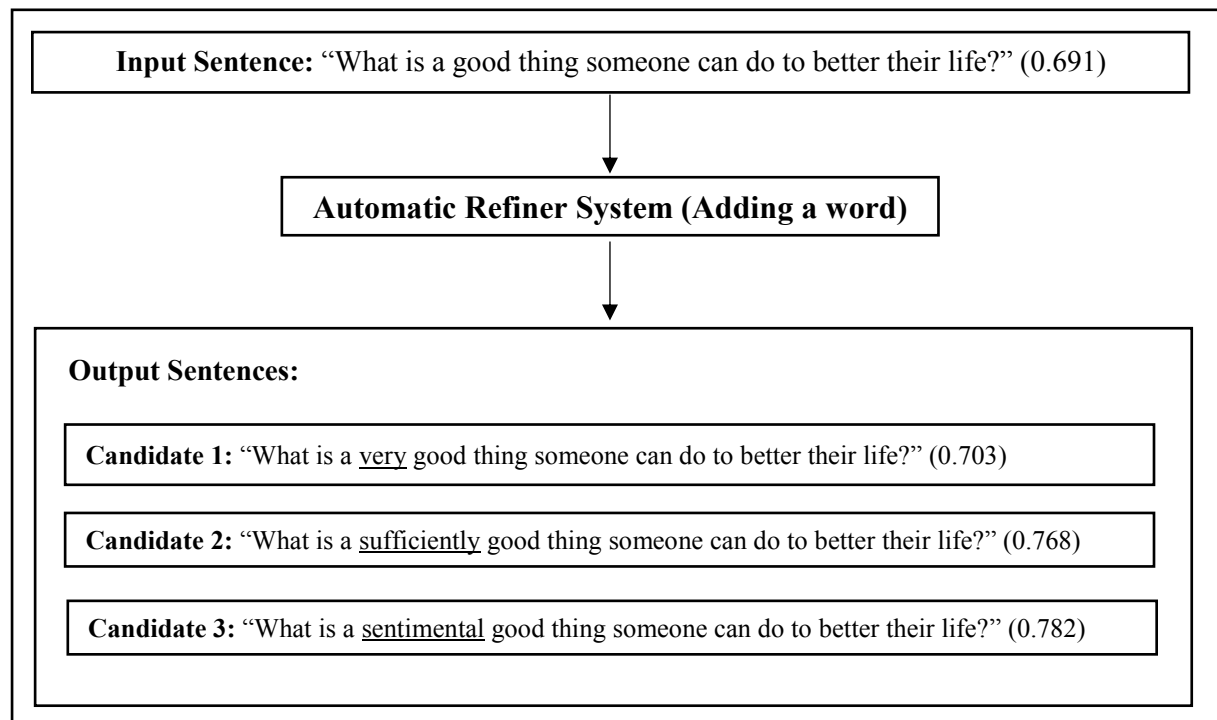


Figure 4.5 Outputs from adding a word to the input sentence “What is a good thing someone can do to better their life?” The words added are underlined.

Three candidate sentences with their predicted probability are shown in Figure 4.5. Underline words are words that automatic refiner system adds to create new sentences with higher probability of becoming a popular submission compared to the input sentence.

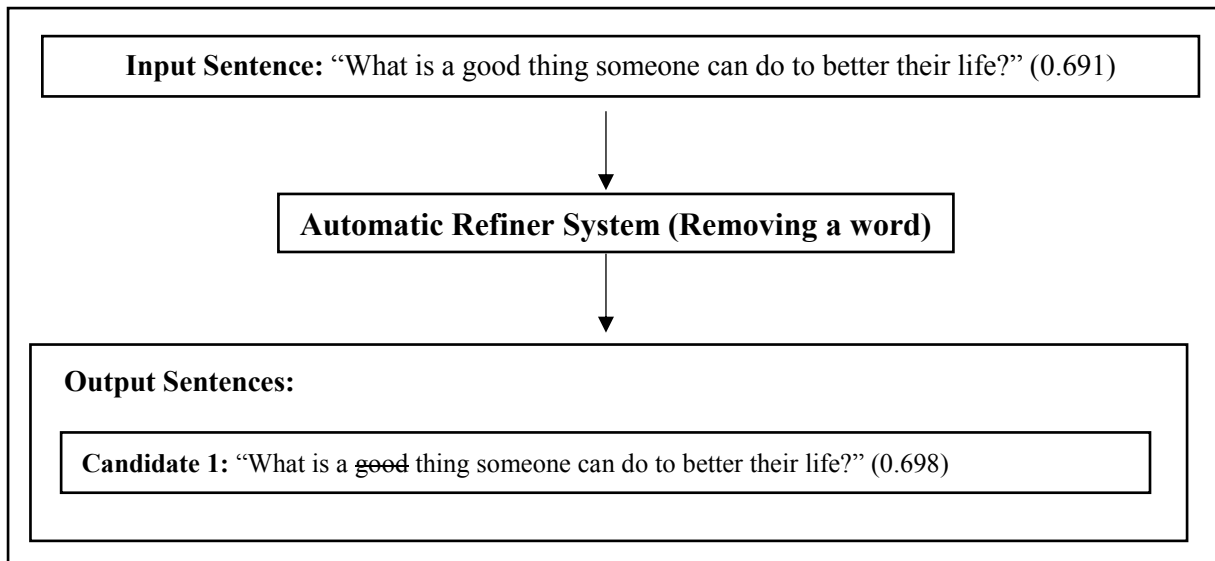


Figure 4.6 Outputs from removing a word from the input sentence “What is a good thing someone can do to better their life?”

From figure 4.6, there is only one candidate sentence. Strikethrough word is a word that automatic refiner system removed to create new a sentence with a higher probability of becoming a popular submission compared to the input sentence.

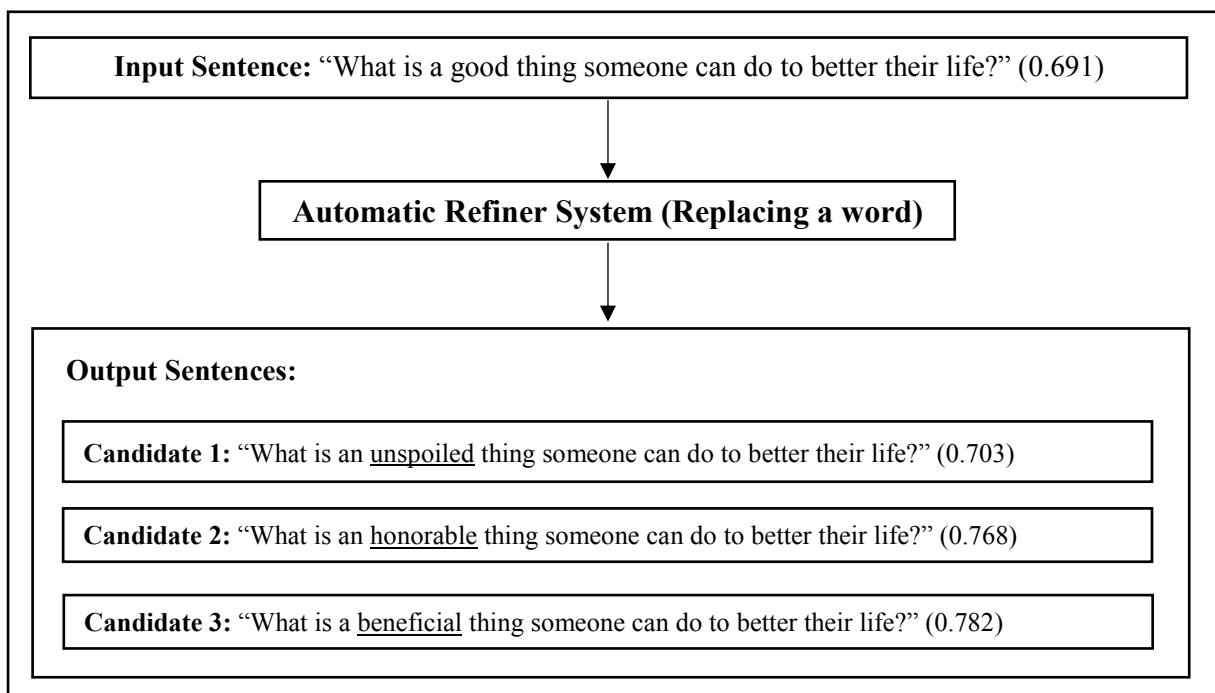


Figure 4.7 Outputs from replacing the word “good” in the input sentence “What is a good thing someone can do to better their life?” with a new word

From figure 4.7, there are three candidate sentences. Underline words are substitute words that automatic refiner system replaced the word “good” to create new sentences with a higher probability of becoming a popular submission compared to the input sentence.

Those new candidate sentences are options that user can later choose for their final submission. However, users can only choose to perform one sentence edition method; adding a word, deleting a word or replacing a word. They cannot choose to perform two or three of those methods at the same time.

4.3 Statistics of generated sentences by the automatic refiner.

We chose 1,000 random AskReddit titles from our dataset and fed them into our automatic refiner system. In average, the submission title was 11.22 words long. For each submission, the automatic refiner system performed the sentence modification methods, a single change at a time. For each method, it outputs a set of candidate sentences. The number of candidate sentences depends on the input sentence and the modification approach.

4.3.1 Single-word insertion

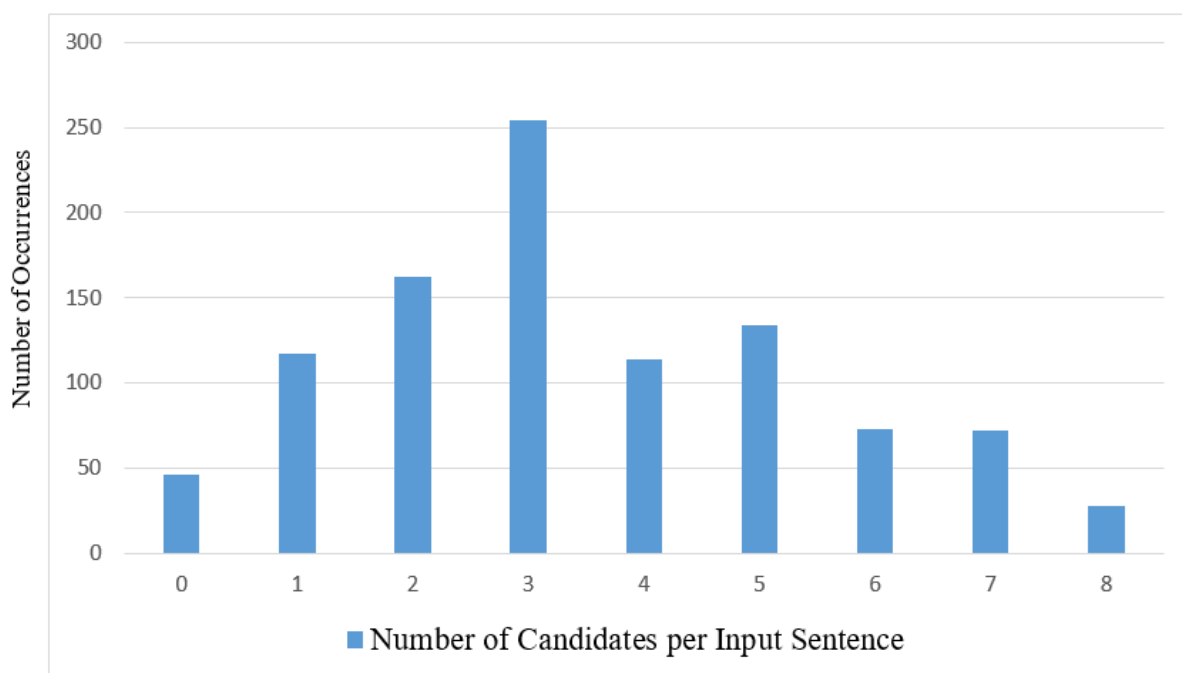


Figure 4.8 The distribution over the number of candidates generated per an input sentence in case of a word being added.

For the method of adding a new word, the number of candidate sentences was varied from zero to eight. The system commonly offered three suggestions.

4.3.2 Single-word deletion

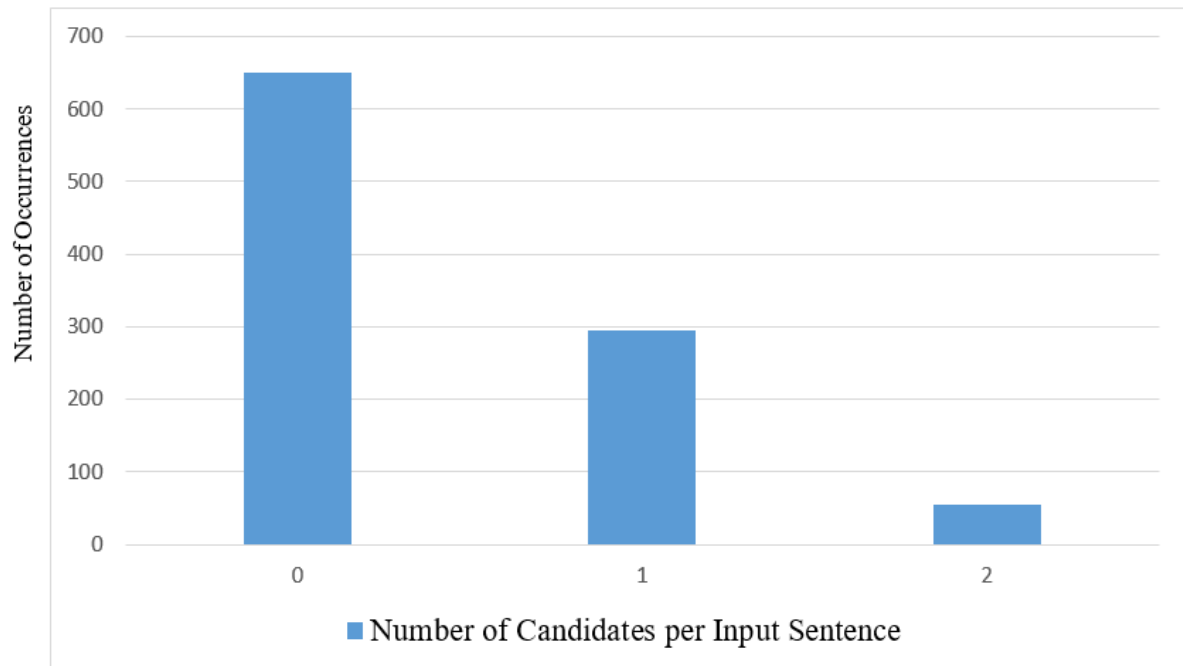


Figure 4.9 The distribution over the number of candidates generated per an input sentence in case of a word being removed.

For a method of removing a word, Approximately 60% of all the input titles, the system gave no suggestion at all, while the maximum number of suggestions with a higher probability of becoming more popular was only two candidate sentences.

4.3.3 Single-word replacement

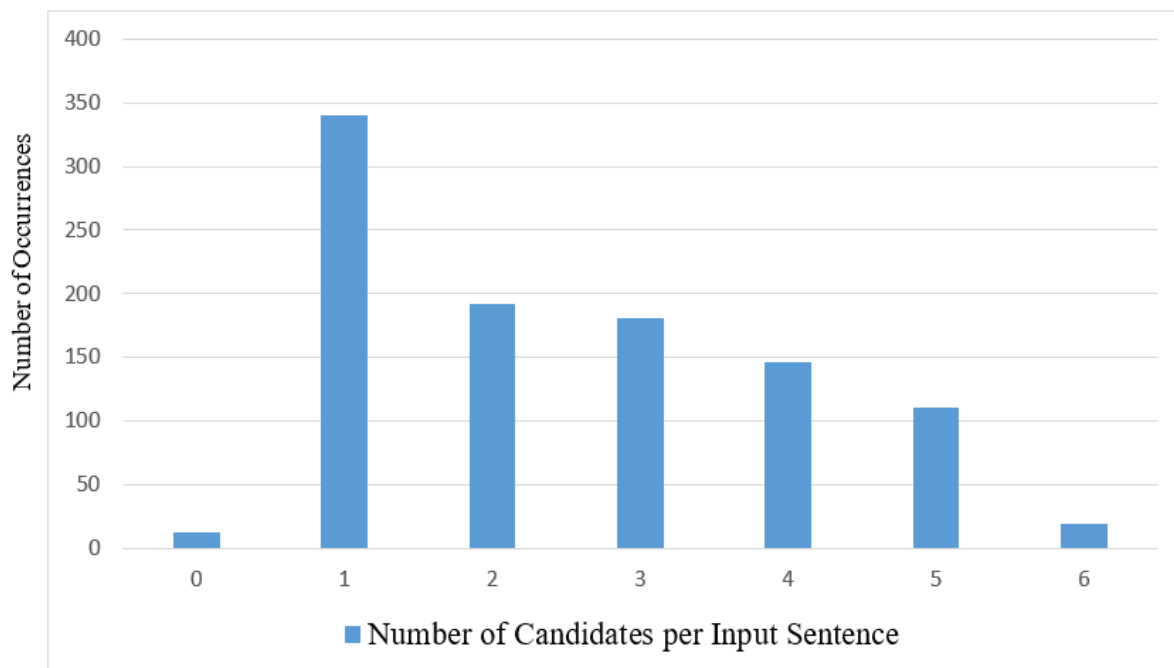


Figure 4.10 The distribution over the number of candidates generated per an input sentence in case of a word being substituted.

For the method of replacing a word with its synonyms, the numbers of candidate sentences were varied from zero to six. Only a candidate sentence was often suggested by the title refiner, whereas few suggestions were given for about a half of the input sentences.

4.4 Human Evaluation

We used a person from the faculty of Arts at Chulalongkorn University to approve the correctness of candidate outputs. A sentence is approved if it is grammatically and semantically correct. We defined a set of candidate sentences resulted from a single change to an input sentence as a successful output if it has at least half of the suggested sentences are approved. For example, a set of three candidates must have at least two approved sentences to be considered a successful output.

Below are the results of human evaluation. “No result” indicates an output with an empty set of candidate sentences.

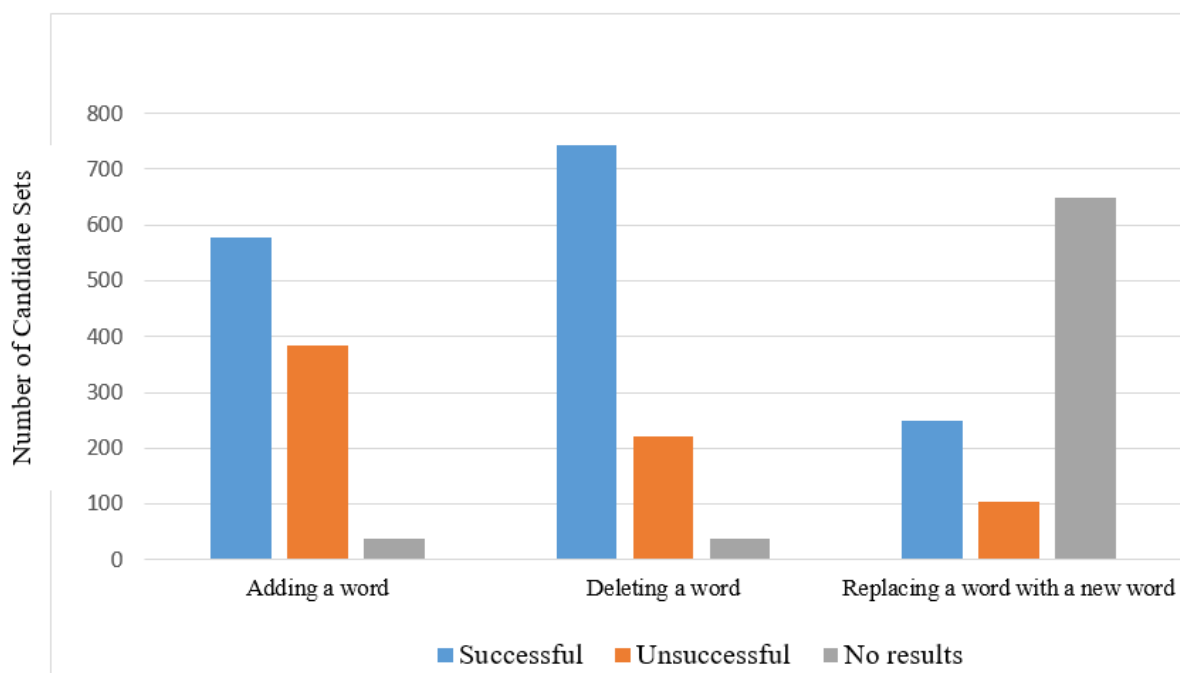


Figure 4.11 The number of successful and unsuccessful candidate sets, categorized by the edition methods.

From figure 4.11, the number of successful candidate sets in each edition methods is higher than the number of unsuccessful candidate sets. However, the number of no results in the method of replacing a word is higher than the number of successful and unsuccessful candidate sets combined.

We also calculated the success rate for each modification approach according to the following Equation (13).

$$Success\ Rate = \frac{Number\ of\ Successful\ Candidate\ Sets}{Number\ of\ Unempty\ Candidate\ Sets} \quad (13)$$

Replacing a word with its synonyms yielded the highest success rate (77.13%), followed by deleting a word (70.66%) and inserting a word (60.16%), respectively.

CHAPTER V

DISCUSSION

5.1 Discussion

We conclude that submissions between 6 AM and 9 AM of the day are more likely to receive better scores than those during other times of the day. Including the information of date and time properties helps the system to make a more accurate prediction on popularity scores.

Our automatic refiner system successfully suggests new candidate sentences from all types of the proposed modification. While word replacement had the highest success rate, word addition had the lowest success rate. That is because replacing a word was done by the concept of synonym so that it did not cause any problems in terms of meaning or part-of-speech. On the other hand, adding a new word to a sentence caused some serious grammatical problems, despite the usage of bigrams. For example, adding an adjective in front of a noun that already has an adjective can cause an “order of adjectives” problem. This is also true for adding an adverb into a sentence.

A handful number of candidate sentences per an input sentence was automatically recommended by the refiner system. However, when the system tried to perform word deletion, more than half of all candidate sets gained less popularity predictively. The main reason is that most of the submission titles we obtained did not contain adjectives nor adverbs. Due to the constraints, we set to avoid breaking the grammar, our automatic refiner system had no choice to remove any word from them. Therefore, it did not make any change to the input sentences and consequently output an empty candidate set instead.

The human evaluation helped evaluate the candidate sentences suggested by our automatic refiner system. Since the system alone does not understand the meaning of each sentence, it cannot semantically incorrect sentences although being grammatically corrected. However, the human evaluation in the project was conducted by a linguistic expert only. In order to enhance the confidence of the candidate modified sentences, one can evaluate based on agreement of a group of people instead.

5.2 Problems and Obstacles

In this research, the train-test split method was used to split the data so that the model was able to be trained in a reasonable amount of time. If we had better computer hardware, we would have considered using a cross-validation method for a better way to report the model performance.

We have tested three grammar checker libraries; GrammarBot, TextBlob and Language-check. GrammarBot had long waiting time between function calls. TextBlob had poor

performance on detecting grammatical errors. Language-check had less waiting time between function calls compared to GrammarBot and was able to detect more errors in a sentence compared to TextBlob. Therefore, we decided to use Language-check in this project. However, it still cannot detect all errors in every sentence. A more suitable grammar checker library should be considered.

Our last problem in this project was to correct the outputs from the automatic refiner. Grammatical errors can be fixed to some extent but semantic errors are one of the most important issues for gaining popularity. We can verify the errors using human evaluation, but we cannot fix them automatically.

5.3 Conclusion and Future Works

In conclusion, our automatic refiner system is able to perform sentence edition and suggest new candidate sentences that are likely to become a more popular submission. However, the score prediction model can be further improved by taking word sequences into account and a more suitable method for semantic evaluation on the modified sentences is necessary for the refiner system.

REFERENCES

- [1] Sze, Vivienne; Chen, Yu-Hsin; Yang, Tien-Ju; Emer, Joel (2017). "Efficient Processing of Deep Neural Networks: A Tutorial and Survey"
- [2] Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg; Dean, Jeffrey (2013). "Distributed Representations of Words and Phrases and their Compositionality".
- [3] Ajitesh, Kumar (2018). "Introduction to Language Model".
- [4] Corbin, Kyle (1989). "Double, Triple, and Quadruple Bigrams". *Word Ways*. **22** (3). Retrieved 11 September 2016.
- [5] Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann. **2** (12): 1137–1143.
- [6] "What is the difference between the test set and validation set?". Retrieved October 10, 2018, from <https://stats.stackexchange.com>
- [7] 80 Amazing Reddit Statistics and Facts. Retrieved December 15, 2018, from <https://expandedramblings.com/index.php/reddit-stats/>
- [8] Top Subreddits. Retrieved June 5, 2018, from <http://redditmetrics.com/top>
- [9] Max Woolf. Predicting the Success of a Reddit Submission with Deep Learning and Keras. Retrieved June 26, 2018, from <https://minimaxir.com/2017/06/reddit-deep-learning/>
- [10] word2vec. Retrieved July 30, 2018, from <https://code.google.com/archive/p/word2vec/>
- [11] GloVe: Global Vectors for Word Representation. Retrieved September 15, 2018, from, <https://nlp.stanford.edu/projects/glove>
- [12] Google. BigQuery Data form Reddit Retrieve December 1, 2018, from https://bigquery.cloud.google.com/table/fh-bigquery:reddit_posts.2018_01
- [13] Francois Chollet. Using pre-trained word embeddings in a Keras model. Retrieve July 16, 2018, from <https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.htm>

APPENDIX

APPENDIX
A The Project Proposal of Course 2301399 Project Proposal
แบบเสนอหัวข้อโครงการ รายวิชา 2301399 Project Proposal
ปีการศึกษา 2561

ชื่อโครงการ (ภาษาไทย)	ระบบทำนายความนิยมของหัวข้อกระทู้ใน AskReddit และตัวช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติ
ชื่อโครงการ (ภาษาอังกฤษ)	AskReddit popularity score prediction system with an automatic title refiner
อาจารย์ที่ปรึกษา	อ. ดร.นฤมล ประทานวณิช
ผู้ดำเนินการ	1. นายขจรยศ คำคุณ เลขประจำตัวนิสิต 5833611023 2. นายภัทรธร ปัทมสังข์ เลขประจำตัวนิสิต 5833650523 สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และ วิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์วิทยาลัย

หลักการและเหตุผล

เว็บบอร์ด เป็นเว็บไซต์ที่ผู้ใช้งานสามารถเข้ามาตั้งหัวข้อ (กระทู้) เพื่อพูดคุย แลกเปลี่ยนความคิดเห็นซึ่งกันและกัน ในเรื่องต่าง ๆ ตัวอย่างของเว็บบอร์ดที่ได้รับความนิยมสูงสุดในประเทศไทยคือ เว็บ pantip และในต่างประเทศคือ เว็บ Reddit

Reddit จัดเป็นเว็บที่ได้รับความนิยมสูงที่สุดเป็นอันดับที่ 18 ของโลก และเป็นเว็บบอร์ดที่มีผู้ใช้งานสูงที่สุดอันดับหนึ่งของโลก ในแต่ละวันจะมีกระทู้ใหม่ไม่ต่ำกว่า 1 ล้านกระทู้ และมีผู้มาตอบกระทู้แต่ละวันไม่ต่ำกว่า 5 ล้านครั้ง จำนวนผู้ใช้งานในปัจจุบันอยู่ที่ 330 ล้านคน [1] จะเห็นได้ว่า Reddit เป็นเว็บไซต์ที่ใหญ่มากแห่งหนึ่ง จึงต้องมีการแบ่งกระทู้ออกเป็นหมวดหมู่ต่าง ๆ เรียกว่า subreddit กระทู้ที่มีเนื้อหาในเรื่องเดียวกัน ควรจะอยู่ใน subreddit ตัวอย่าง subreddit ได้แก่ การเมือง (politics) การออกกำลังกาย (fitness) และ ตั้งคำถาม (AskReddit) เป็นต้น นอกจากนี้ Reddit ยังมีระบบลงคะแนนเสียงกระทู้ โดยผู้ใช้งานสามารถให้คะแนนกระทู้ที่ผู้ใช้คนอื่นตั้งได้กระทู้ละ 1 ครั้ง กระทู้ที่ได้รับคะแนนมากจะถูกส่งขึ้นไปปรากฏอยู่บนหน้าหลักของเว็บไซต์ ส่งผลให้ผู้ใช้คนอื่น ๆ สามารถเห็นกระทู้ได้ง่ายขึ้น และเมื่อกระทู้เหล่านี้มีผู้ใช้เข้ามามากขึ้น จะส่งผลให้จำนวนความคิดเห็นหรือคอมเมนต์ ภายในกระทู้ เพิ่มสูงมากขึ้นด้วย ทำให้เจ้าของกระทู้และผู้อ่านได้ความรู้ที่เกี่ยวกับกระทู้ดังกล่าวเพิ่มขึ้น

คะแนนความนิยมของกระทู้ขึ้นกับหลายปัจจัย ไม่ว่าจะเป็น หัวกระทู้ เนื้อหากระทู้ หรือรูปภาพประกอบกระทู้ ซึ่งอาจยากต่อการวิเคราะห์ ทางผู้จัดทำโครงการเล่มนี้จึงเลือกศึกษาเฉพาะกระทู้ที่อยู่ใน subreddit ชื่อ AskReddit ซึ่งมีแต่หัวข้อกระทู้ที่เป็นข้อความเท่านั้น เพื่อลดความซับซ้อนของปัจจัยอื่น ๆ ที่อาจจะมีผลต่อความนิยมลง โดยผู้จัดทำตั้งข้อสังเกตว่าคุณสมบัติบางประการที่แฝงอยู่ในรูปประโยคของหัวข้อกระทู้ มีผลทำให้ผู้ใช้งานสนใจ และเข้ามาตอบกระทู้ นอกจากนี้ AskReddit ยังเป็น subreddit ที่มีจำนวนสมาชิกสูงที่สุด 3 อันดับแรกเทียบกับ subreddit อื่น ๆ ดังภาพที่ 1 จึงมั่นใจได้ว่าจะมีข้อมูลที่ใช้ในการวิเคราะห์เพียงพอ อีกทั้งรูปแบบประโยคเป็นคำถามโดยตรงไปตรงมา จึงไม่ต้องกังวลเรื่องความหมายแฝงอื่น ๆ ที่อาจซ่อนอยู่ในประโยค ที่ยากต่อการวิเคราะห์

Rank	Reddit	Subscribers
1	/r/announcements	21,352,277
2	/r/funny	18,974,028
3	/r/AskReddit	18,752,990
4	/r/todayilearned	18,434,007
5	/r/science	18,321,396
6	/r/worldnews	18,310,545
7	/r/pics	18,259,401
8	/r/IAmA	17,745,577
9	/r/gaming	17,696,844
10	/r/videos	17,424,011

ภาพที่ 1 แสดง subreddit ที่มีจำนวนสมาชิกสูงที่สุด 10 อันดับแรก โดย AskReddit อยู่อันดับที่ 3 [2]

จากที่กล่าวมาข้างต้นนี้ ผู้จัดทำจึงคิดพัฒนาระบบทำนายความนิยมของหัวข้อกระทู้ใน AskReddit และตัวช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติเพื่อเพิ่มความนิยมในเว็บ Reddit ใน subreddit ชื่อ AskReddit โดยการแก้ไขประโยคหัวข้อกระทู้ จะใช้วิธีการ เพิ่มคำ ลดคำ หรือเปลี่ยนคำจำนวน 1 คำ ในประโยค การทำงานทั้งหมดนี้จะอาศัยความรู้ในด้านการเรียนรู้ของเครื่อง (machine learning) และ การเรียนรู้เชิงลึก (deep learning) เป็นหลัก โดยหวังว่าเมื่อผู้ใช้งานใช้ระบบที่จะพัฒนานี้เป็นตัวช่วยในการตั้งกระทู้ใหม่ใน AskReddit แล้วจะทำให้กระทู้ที่ตั้งนั้นมีโอกาสได้รับความนิยมสูงกว่าการไม่ใช้ระบบช่วย

งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่ศึกษาโดย Max Woolf [3] ได้วิเคราะห์ว่าปัจจัยใดมีผลกับคะแนนความนิยมบ้าง ปัจจัยหลัก ๆ 3 ปัจจัยที่มีผลต่อคะแนนคือ หัวข้อ (title) เวลา (time) และ เนื้อหา (content) ยกตัวอย่างเช่นในหัวข้อ การออกกำลังกาย (fitness) กระทู้ที่มีคำหรือเนื้อหาเกี่ยวกับการเปลี่ยนแปลงของร่างกาย (transformation) จะได้รับความนิยมที่สูง และถ้าตั้งกระทู้ในช่วงเวลา 8-11 น. ก็จะได้ความนิยมที่สูงเช่นกัน ในการดำเนินงานผู้วิจัยได้ใช้ชุดข้อมูลจาก Google BigQuery และเลือก หัวข้อย่อย (subreddit) ชื่อ AskReddit ซึ่งมีจำนวนกระทู้ใหม่ในแต่ละวันสูงที่สุด เทียบกับ subreddit อื่น ๆ ผู้วิจัยไม่ได้ทำนายคะแนนความนิยมเป็นตัวเลขเพราะคะแนนความนิยมของกระทู้มีค่าการกระจายตัวมากเกินไป จึงใช้วิธีแบ่งกระทู้ออกเป็น 2 กลุ่ม คือ กระทู้ที่ได้รับความนิยม กับ กระทู้ที่ไม่ได้รับความนิยม โดยจะได้ว่าถ้าเปอร์เซ็นต์โทลล์ของคะแนนกระทู้ที่เท่ากับหรือมากกว่า 50 แสดงว่ากระทู้นั้นได้รับความนิยม จากข้อมูลทั้งหมด subreddit fitness มีกระทู้ทั้งหมด 976,538 กระทู้ แบ่งเป็นกระทู้ที่ได้รับความนิยม 36% จากทั้งหมด และอีก 64% คือกระทู้ที่ไม่ได้รับความนิยม แบบจำลอง (model) ที่ใช้ทำนายว่ากระทู้นี้ได้รับความนิยมหรือไม่ได้รับความนิยมคือ logistic regression โดยผู้วิจัยเริ่มจากการทำ word embedding โดยใช้ Glove หลังจากนั้นนำผลลัพธ์ที่ได้ผ่าน fully-connected layers ผลลัพธ์จะบอกได้ว่าข้อความเป็นกระทู้ที่ได้รับความนิยมหรือไม่ โดยกำหนดว่ากระทู้ที่ได้คะแนนที่ทำนายจากแบบจำลองตั้งแต่ 2 คะแนนขึ้นไป จัดเป็นกระทู้ที่ได้รับความนิยม ซึ่งมีความแม่นยำในการทำนาย (accuracy) อยู่ที่ 66.9% นอกจากนี้ ผู้วิจัยทดลองเพิ่ม ลด หรือเปลี่ยนคำ ตั้งแต่ 1 คำขึ้นไปในประโยคหัวข้อกระทู้ด้วยตัวเอง พบว่าการเพิ่ม ลด หรือเปลี่ยนคำ บางคำ ทำให้กระทู้มีโอกาสได้คะแนนสูงขึ้น ผู้จัดทำเห็นการทดลองเหล่านี้แล้ว จึงอยากพัฒนาระบบช่วยปรับข้อกระทู้เพื่อเพิ่มความนิยมในเว็บ Reddit ขึ้นมา ซึ่งการเพิ่ม ลด หรือเปลี่ยนคำ ของระบบนี้ จะทำอย่างอัตโนมัติโดยไม่ต้องใช้คนช่วย

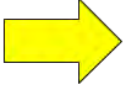
ความรู้ที่เกี่ยวข้อง

1. การดึงข้อมูลและจัดเก็บข้อมูล

ข้อมูลที่จะนำมาใช้คือ หัวข้อกระทู้ คะแนนของกระทู้ และเวลาที่ตั้งกระทู้ จาก www.reddit.com/r/AskReddit การดึงข้อมูลสามารถทำได้ 2 วิธี วิธีแรกคือใช้เครื่องมือที่เว็บ Reddit เตรียมไว้ให้ชื่อว่า Reddit API แต่ด้วยข้อจำกัดของ API ทำให้สามารถดึงข้อมูลได้แค่ครั้งละ 1000 กระทู้ วิธีที่ 2 คือใช้ Google BigQuery ซึ่งเป็นบริการของ Google ใช้สำหรับค้นหาและดึงข้อมูลจากฐานข้อมูลของเว็บไซต์ต่าง ๆ วิธีนี้สามารถดึงข้อมูลจากฐานข้อมูลของ Reddit ได้ครบถ้วนและไม่มีข้อจำกัดในเรื่องจำนวนกระทู้ ข้อมูลที่ถูกดึงมาจะถูกจัดเก็บอยู่บน Google Cloud ผู้จัดทำเลือกใช้วิธีที่ 2 เพราะมีความสะดวกในการใช้งาน และง่ายต่อการจัดเก็บข้อมูล

2. การประมวลผลข้อความ (Text Processing)

คอมพิวเตอร์ไม่สามารถประมวลผลข้อความได้อย่างมนุษย์ จึงต้องมีการแปลงข้อความให้อยู่ในรูปตัวเลขเพื่อให้คอมพิวเตอร์สามารถประมวลผลได้ วิธีการที่ง่ายที่สุดในการแปลงข้อความให้เป็นตัวเลข คือการใช้ One-hot Encoding ซึ่งจะแปลงคำแต่ละคำให้อยู่ในรูปของไบนารีเวกเตอร์ (binary vector) ตัวอย่างเช่น มีคำศัพท์ทั้งหมด 3 คำปรากฏอยู่ในข้อความ คือ Red Yellow และ Green การแปลงแบบ One-hot encoding จะทำได้ตามภาพที่ 2



Color	Red	Yellow	Green
Red	1	0	0
Red	1	0	0
Yellow	0	1	0
Green	0	0	1
Yellow	0	0	1

ภาพที่ 2 วิธีการแปลงแบบ One-hot encoding

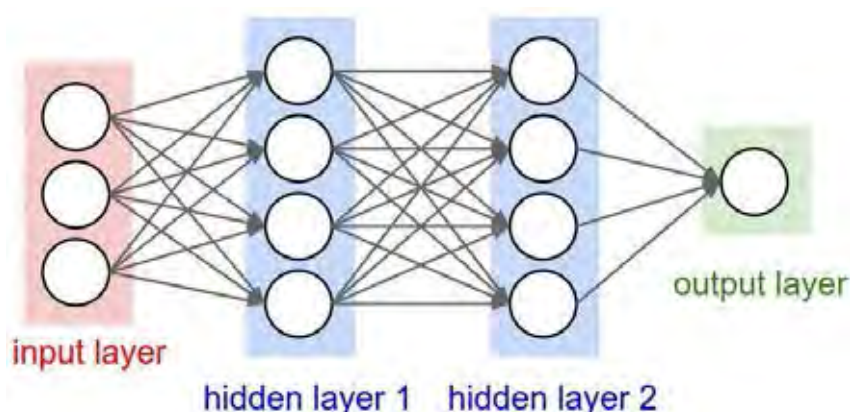
แต่หากจำนวนคำศัพท์ทั้งหมดที่ปรากฏในชุดข้อมูลมีจำนวนมาก ขนาดของเวกเตอร์ที่ใช้แทนคำแต่ละคำก็จะเพิ่มมากขึ้น ส่งผลให้เนื้อที่ทั้งหมดที่ต้องใช้เพิ่มสูงขึ้น และต้องใช้เวลาประมวลผลมากขึ้นนั่นเอง จึงได้มีการคิดค้นวิธีอื่น ๆ เพื่อแปลงข้อความให้อยู่ในรูปตัวเลข และลดขนาดเวกเตอร์ลง วิธีการนี้เรียกว่า word embedding และวิธีที่นิยมกันในปัจจุบัน คือ word2vec

word2vec เป็นการทำ word embedding แบบหนึ่ง คิดค้นโดย Google [4] กล่าวคือ คำแต่ละคำ จะมีเวกเตอร์เป็นอย่างไร ขึ้นอยู่กับบริบท (context) ของคำนั้น ๆ กับข้อความรอบข้าง ตัวอย่างเช่น คำว่า คนแก่ กับ คนชรา ใช้ในบริบทที่คล้ายกัน word2vec จะสามารถเรียนรู้และแยกแยะได้เองว่า คนแก่ กับ คนชรา ใช้ในความหมายเหมือน ๆ กัน เป็นต้น นอกจากนี้ ผลของการทำ word embedding ด้วยการใช้ word2vec ทำให้สามารถวัดความเหมือนระหว่างคำได้ (word similarity)

GloVe เป็นการทำ word embedding อีกรูปแบบหนึ่ง คิดค้นโดย Stanford [5] ทั้ง 2 วิธีมีประสิทธิภาพใกล้เคียงกัน แต่ GloVe สามารถเรียนรู้คำได้รวดเร็วกว่า word2vec ในขณะเดียวกันก็ใช้เนื้อที่ในการจัดเก็บมากกว่า ทั้ง 2 วิธีมีแบบจำลองที่ถูกลงสอนไว้แล้ว (pre-trained models) ในที่นี้ผู้จัดทำโครงการจะขอเลือกใช้ GloVe และใช้ pre-trained model ที่มีอยู่แล้วในการทำ word embedding หัวข้อกระทู้

3. โครงข่ายประสาทเทียมเชิงลึก (Deep neural network)

Deep learning เป็นศาสตร์แขนงหนึ่งของ Machine learning ซึ่งเลียนแบบเซลล์ประสาทของมนุษย์ (Neural networks) โดยการจำลองเซลล์ประสาทของมนุษย์ในระบบคอมพิวเตอร์ เรียกว่า โครงข่ายประสาทเทียม (Artificial neural network) ประกอบด้วย โหนด (node) และเส้นเชื่อมต่าง ๆ พร้อมกับค่าน้ำหนัก (weight) ส่วนการทำงานของโครงข่ายประสาทเทียม มีอยู่ด้วยกัน 3 ส่วนคือ ส่วนชั้นข้อมูลนำเข้า (input) ส่วนชั้นซ่อนตัว (hidden layer) และส่วนชั้นข้อมูลนำออก (output) เราเรียกโครงข่ายประสาทเทียมที่มี hidden layer มากกว่า 1 ชั้นว่า โครงข่ายประสาทเทียมเชิงลึก ด้วยสถาปัตยกรรมแบบนี้ทำให้โครงข่ายประสาทเทียมเชิงลึกสามารถเรียนรู้ความสัมพันธ์ระหว่างข้อมูลนำเข้าและข้อมูลนำออกที่ซับซ้อนได้



ภาพที่ 2 ตัวอย่าง Deep Neural Network ที่มี hidden layer 2 ชั้น

วัตถุประสงค์

เพื่อพัฒนาระบบทำนายคะแนนความนิยมของหัวข้อกระทู้ใน AskReddit และตัวช่วยปรับแต่งหัวข้อกระทู้อัตโนมัติเพื่อเพิ่มโอกาสที่หัวข้อกระทู้จะได้รับคามนิยม โดยการเพิ่มคำ ตัดคำออก หรือเปลี่ยนคำ ในหัวข้อกระทู้จำนวน 1 คำ

ขอบเขตของโครงการ

1. ใช้ subreddit ที่มีแต่หัวข้อเพียงอย่างเดียว ไม่มีเนื้อหาในกระทู้ ในที่นี้จะใช้ subreddit ที่ชื่อว่า AskReddit
2. ข้อความที่เป็นหัวข้อที่ใช้ในการศึกษาและพัฒนาต้องเป็นภาษาอังกฤษเท่านั้น
3. การแนะนำในการตั้งกระทู้จะสนใจแค่การเพิ่มคำ ตัดออก และเปลี่ยนคำจำนวน 1 คำเท่านั้น

วิธีการดำเนินงาน

1. ศึกษางานเนื้อหาและบทความเกี่ยวกับการทำนายความนิยมจากหัวข้อกระทู้ Reddit โดยใช้โครงข่ายประสาทเทียมเชิงลึกซึ่งการฝึกสอน (training) แบบจำลองจะเป็นแบบ supervised learning
2. เก็บข้อมูลหัวข้อกระทู้ เวลาที่ตั้งกระทู้ และคะแนนความนิยมจาก Google BigQuery [6]
3. พัฒนาแบบจำลองโดยใช้ Deep Neural Networks
 - 3.1. ทำ word embedding กับข้อความจากหัวข้อกระทู้ ส่วนเวลาแบ่งได้เป็น นาที วันในสัปดาห์ และ ปี ที่ตั้งกระทู้ทำเป็น one-hot encoding เพื่อพัฒนาแบบจำลองที่จะใช้ทำนายคะแนนนิยมจากเวกเตอร์ข้างต้น
 - 3.2. ทำแบบจำลองเพื่อ เพิ่ม ลด หรือปรับเปลี่ยนข้อความในประโยค เพื่อให้ได้ประโยคใหม่ที่ได้คะแนนความนิยมสูงกว่าเดิม
4. พัฒนาและทดสอบความถูกต้องของแบบจำลอง
5. ทดสอบวัดประสิทธิภาพของแบบจำลองที่ได้
 - 5.1. แบ่งข้อมูล 80% ของทั้งหมด สำหรับข้อมูลฝึกสอน (training set) และอีก 20% ของทั้งหมดสำหรับข้อมูลทดสอบ (testing set) และวัดผลการทำนายความนิยมบนข้อมูลทดสอบ โดยหากแบบจำลองที่ได้ มีความแม่นยำในการทำนายความนิยมเกินกว่า baseline ที่ใช้หลัก majority จะถือว่าเป็นแบบจำลองที่มีประสิทธิภาพ โดยหลัก majority นิยมใช้เป็น baseline สำหรับปัญหาการจัดข้อมูลออกเป็นกลุ่ม (classification) ซึ่งสอดคล้องกับปัญหาในโครงการนี้คือการจัดกลุ่มกระทู้ที่ได้รับความนิยมและกระทู้ที่ไม่ได้รับความนิยม โดยหลักการนี้จะจัดกลุ่มข้อมูลทุกตัวให้อยู่ในกลุ่มข้อมูลข้างมาก เช่น ข้อมูลกลุ่มแรกมีจำนวน 70% กลุ่มที่สอง 30% ถ้าใช้หลัก majority ผลลัพธ์จะมีความแม่นยำอยู่ที่ 70% เป็นต้น
 - 5.2. วัดผลการปรับปรุงประโยค โดยเริ่มจากการแก้ไขคำในประโยคก่อน ด้วยการเพิ่มคำ ตัดคำ และเปลี่ยนคำ โดยทำแต่ละวิธีอย่างน้อย 1 ครั้ง เพื่อดูว่าประโยคใหม่ที่ได้มีค่าความน่าจะเป็นที่จะได้รับความนิยม สูงขึ้น จากนั้นจึงทำการแก้ไขไวยากรณ์ของประโยคให้ถูกต้อง
6. ปรับปรุงแบบจำลองเพื่อเพิ่มประสิทธิภาพ
7. สรุปผลการดำเนินงาน ข้อเสนอแนะและการจัดทำเอกสาร

ตารางเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน	ปี 2561					ปี 2562			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. ศึกษางานเนื้อหาและบทความเกี่ยวกับการทำนายความนิยมจากหัวข้อกระทู้ reddit									
2. กำหนดชุดข้อมูลที่ใช้ในการฝึกสอนและทดสอบ									
3. วิเคราะห์พีเจอร์และออกแบบสถาปัตยกรรมโครงข่ายประสาทเทียม									
4. พัฒนาและทดสอบความถูกต้องของโมเดล									
5. ทดสอบวัดประสิทธิภาพของวิธีการที่นำเสนอ									
6. ปรับปรุงเพื่อเพิ่มประสิทธิภาพ									
7. สรุปผลการดำเนินงาน ข้อเสนอแนะและการจัดทำเอกสาร									

ประโยชน์ที่เป็นเหตุผลในการพัฒนาโปรแกรม

1. ประโยชน์ในด้านความรู้และประสบการณ์ต่ออนิสิตเอง
 - 1.1. ได้รับประสบการณ์การทำงานเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) แบบต่าง ๆ รวมถึงการเรียนรู้เชิงลึก (Deep learning)
 - 1.2. ได้ความรู้เกี่ยวกับ Natural Language Processing (NLP)
 - 1.3. เข้าใจการทำงานอย่างเป็นระบบและการทำงานเป็นทีม
 - 1.4. มีความรู้ ความเข้าใจในการทำระบบแนะนำ
 - 1.5. ได้รับประสบการณ์การใช้งาน cloud computing และการทำ API

2. ประโยชน์ต่อผู้ใช้งานและสังคม
 - 2.1. ได้นำเสนอวิธีการใหม่ที่ต่อยอดมาจากงานวิจัยในอดีต ซึ่งอาจเป็นประโยชน์ต่อในวงการวิจัยที่เกี่ยวข้องกับการนำเสนอประโยค
 - 2.2. ผู้ใช้ได้รับความรู้เกี่ยวกับองค์ประกอบของประโยคที่ใช้เป็นหัวข้อซึ่งทำให้กระทู้ได้รับความนิยม
 - 2.3. หากผู้ใช้ต้องการจะตั้งกระทู้ใหม่ใน Reddit ก็สามารถใช้ระบบนี้เพื่อทำนายความนิยมก่อนได้ ทำให้ผู้ใช้ได้รับหัวข้อที่ทำให้กระทู้มีโอกาสได้รับความนิยมสูง
 - 2.4. สามารถใช้เพื่อการตลาดได้

อุปกรณ์และเครื่องมือที่ใช้

1. ฮาร์ดแวร์ (Hardware)
 1. Macbook Pro 2017
 2. Notebook CPU Intel(R) Core(TM) i7-7500U ความเร็ว 2.70GHz 2.90 GHz RAM 8.00 GB
2. ซอฟต์แวร์ (Software)
 1. Python 3.6
 2. Google BigQuery
 3. Google Cloud Platform
 4. Google API service
 5. Machine learning libraries Keras [7]

งบประมาณ

- | | | |
|---|------|-----|
| 1. ค่าเช่า Google Cloud Platform | 9500 | บาท |
| 2. กระดาษถ่ายเอกสาร A4 80 แกรม และหมึกพิมพ์ | 500 | บาท |

รวม 10000 บาท

BIOGRAPHY



Mr. Pattaratorn Pattamangsang

Department of Mathematics and Computer Science

Faculty of Science, Chulalongkorn University

Email: pattamue@gmail.com



Mr. Khajornyot Khamkhon

Department of Mathematics and Computer Science

Faculty of Science, Chulalongkorn University

Email: benzpos@gmail.com