USE OF WORD EMBEDDING TO GENERATE SIMILAR WORDS AND MISSPELLINGS FOR TRAINING PURPOSE IN CHATBOT DEVELOPMENT

by

SANJAY THAPA

THESIS

Submitted in partial fulfillment of the
requirements for the degree of Master of
Science in Computer Science at
The University of Texas at Arlington
December, 2019

Arlington, Texas

Supervising Committee:

Deokgun Park, Supervising Professor
Manfred Huber
Vassilis Athitsos

ACKNOWLEDGEMENTS

I would like to thank Dr. Deokgun Park for allowing me to work and conduct the research in the Human Data Interaction (HDI) Lab in the College of Engineering at the University of Texas at Arlington. Dr. Park's guidance on the procedure to solve problems using different approaches has helped me to grow personally and intellectually. I am also very thankful to Dr. Manfred Huber and Dr. Vassilis Athitsos for their constant guidance and support in my research. I would like to thank all the members of the HDI Lab for their generosity and company during my time in the lab. I also would like to thank Peace Ossom Williamson, the director of Research Data Services at the library of the University of Texas at Arlington (UTA) for giving me the opportunity to work as a Graduate Research Assistant (GRA) in the dataCAVE.

DEDICATION

I would like to dedicate my thesis especially to my mom and dad who have always been very supportive of me with my educational and personal endeavors. Furthermore, my sister and my brother played an indispensable role to provide emotional and other supports during my graduate school and research.

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABSTRACT

USE OF WORD EMBEDDING

TO GENERATE SIMILAR WORDS AND MISSPELLINGS FOR

TRAINING PURPOSE IN CHATBOT DEVELOPMENT

Sanjay Thapa, M.S.

The University of Texas at Arlington, 2019

Supervising Professor: Deokgun Park

The advancement in the field of Natural Language Processing and Machine Learning has played a significant role in the huge improvement of conversational Artificial Intelligence (AI). The use of text-based conversation AI such as chatbots have increased significantly for the everyday purpose to communicate with real people for a variety of tasks. Chatbots are deployed in almost all popular messaging platforms and channels. The rise of chatbot development frameworks based on machine learning is helping to deploy chatbot easily and promptly. These chatbot development frameworks use machine learning and natural language understanding (NLU) to understand users' messages and intents and respond accordingly to users' utterance. Since most of the chatbots are developed for domain-specific purposes, the performance of the chatbot is directly related to the training data. To increase the domain knowledge and knowledge base of the chatbots via training data, the chatbots need to know similar words or phrases for a

users' message. Furthermore, it is not guaranteed that a user will spell a word correctly. A lot of times, in written conversation, a user will misspell at least some words. Thus, to include semantically similar words and misspellings in the training data, I have used word embedding to generate misspellings and similar words. These generated similar words and misspellings will be used as training data to train the model for chatbot development.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Different methods and techniques are used by computers to understand human language. These techniques, in general, used by computer programs to interpret human language are referred to as Natural Language Processing (NLP) [6]. Jacob Eisenstein has succinctly described NLP as "methods for building computer software that understands, generates, and manipulates human language " [6]. One of the subfields of NLP is Natural Language Understanding (NLU). NLU is used for relational extraction, paraphrase and natural language inference, semantic parsing, sentiment analysis, dialogue agents, summarization, and question answering [20].

The recent developments in the field of applied machine learning have played a key role in the improvement in methods used for NLP and NLU. Because of the availability of a huge amount of data from different sources like Facebook, Twitter, Wikipedia, and other easily and readily available sources, researchers and computer scientists have the advantage of experimenting with different sets of data. Furthermore, the improvement and the accessibility of the computing infrastructure and processing power like Graphical Processing Unit (GPU) has drastically reduced the computing time for researchers allowing them to conduct research and experiments within hours instead of days. Also, machine learning approaches have resulted in state-of-the-art results in different fields of Artificial Intelligence [12]. With the selection of the right set of training data and the algorithms, NLU now can correctly predict the intent of the user's utterance or sentence. The intent of a user may be defined by a single word, phrase, or sentence.

In chatbots or any form of written communications, it is not guaranteed that a person will ask or reply to the chatbot in complete sentences. A user may use synonyms or similar phrases

for the same intent or goal. Also, while a user types a message, the user may misspell a word or words. The chatbot system needs to handle such misspellings to reply to the user with the correct response. Handling misspellings and similar words will help to maintain good communication between the user and the chatbot. The ability to handle similar phrases and misspellings is of uttermost importance for the satisfaction of the user and the performance of the chatbot system.

In this thesis, I will generate similar words to a word using the word embedding technique. Initially, I will generate some possible misspellings using an edit distance of 1 for 10 words by considering some assumptions and findings from previous research. While doing so, I will be using only misspellings which are possible by using insertion, deletion, and substitution techniques. This approach of using an edit distance of 1 may result in many misspellings that may never happen in real-world communication. To eliminate the misspellings that will never happen in the real text-based conversation, I will use the combination of word embedding and edit distance of 1 to find the misspellings that have happened in Twitter [9][17][18]. Those misspellings are included in the training dataset of the chatbot to increase its domain knowledge. Similarly, 30 most similar words from the word embedding model using the similarity measure cosine similarity is used to find similar words for the selected 10 words. All of the 30 most similar words will not be included in the training data as some of the similar words for the given word may belong to the other intents or may not be relevant for the training purpose.

In chapter 2, I will briefly discuss the conversation AI, the architecture of Rasa: a machine learning framework for chatbot development [2]. Chapter 3 will explain the edit distance, and why edit distance or Levenshtein distance approach is not feasible for the generation of misspellings by itself. And, finally, I will generate similar words for the given word which can be used in the training data of chatbot.

Chapter 5 includes the results and findings of the misspellings and similar words for the given words using the combination of word embedding and edit distance of 1.

CHAPTER 2

RASA: A CHATBOT FRAMEWORK BASED ON MACHINE LEARNING

2.1. Related Works

A chatbot is a computer program that can interact with a user using a message or speech in a conversational form. One of the earliest versions of the chatbot was developed at the Massachusetts Institute of Technology (MIT) Artificial Intelligence Laboratory. The name of the chatbot was ELIZA [7], designed and developed by Joseph Weizenbaum in the mid-1960 [22]. This chatbot used "pattern matching and substitution methodology" [7] to present the response to the user. This chatbot was one of the earliest chatbots incorporating natural language processing.

The abundance of data, computing power, and the recent development in the field of NLP has slowly shifted the design of the chatbot from task-oriented to the data-driven conversational chatbots [4]. Some of the recent popular chatbot frameworks based on machine learning are Microsoft Bot Framework by Microsoft, Dialogflow by Google, IBM Watson by IBM, Amazon Lex by Amazon, and Rasa by Rasa Technologies Inc. For the research, I chose to use the Rasa framework as this is an open-source framework. Furthermore, by using Rasa, the user can take full control of the data as the user does not have to send their data to the third-party servers. The other beneficial feature of using rasa is that once the chatbot is made, it can be easily deployed in other messaging platforms like Facebook Messenger, Slack, and other messaging platforms.

2.2. **Conversational Artificial Intelligence (AI)**

Conversation AI is used to automate communication either in a written or spoken form to create a scalable personal experience. The forms of Conversational AI can be different. It may be

in the form of speech-based assistants like Alexa, or Siri, or text-based platforms or chatbots [3]. Some of the common terms used in the field of Conversational AI are:

2.2.1. Utterances:

The input from the user to the system. The utterance can be a single character, a word or more words, letters, one sentence, and or more than one sentence. For instance, when a user type "Y" or "N", it is considered as utterance. Also, the sentences like "I want to order vanilla ice cream" can be the next utterance of the user.

2.2.2. Intents:

The intention of the users or the purpose of a user's message is known as intent. In other words, intent can be regarded as a summary or the goal of the users. For a user's utterance, "I want to order vanilla ice cream", the intent can be 'order_ice_cream.' While having a conversation in both written and spoken form, there are so many "Unwritten rules of communication" [1]. For instance, a conversation between a customer in a restaurant and the waitress can occur in so many ways. For instance, when a waitress asks a customer for a drink: "What would you like to order for a drink?" This is a complete sentence from the waitress's side, but a customer may reply in different ways. For instance, s/he may say "No", "I want water", "I want coke", "I want to order Sprite" "I would like to get Diet Coke". This is a very simple conversation, and you can see from the above examples that written and spoken conversation can happen using a simple word "No" to the combination of words. This is simply a way human beings convey their meaning or intent to others. Effective conversation only occurs when both sides can understand each other. But the understanding of the message also depends on the knowledge level of the people. For instance, the common synonyms of the word “beautiful” can be attractive, pretty, gorgeous, or nice-looking. Even though these words are synonyms, the

choice of the words matters while using the word. One of the great examples in the use of synonyms word is the use of the word “frantic” and “breadth-taking.” Those words are listed as synonyms, but the use of the word depends on the context, place, and scenario. The wrong choice of the word will result in poor communication.

2.2.3. Entities and Slots:

Entities are specific and detailed information provided by the user. In our example of the utterance, "I want to order vanilla ice cream," vanilla will be an entity. The next customer or a user may want to order chocolate ice cream, so he may say "I want to order chocolate ice cream." Thus, in these two distance, the overall intent of the users will be the same, but the type of ice cream will be different, hence vanilla and chocolate will be entities and slots for the utterance. To process the intent of the user, we need to save the entities in the memory of the dialogue management.

2.2.4. Response

The response is simply a reply from the chatbot or voice-based assistants. The response from these systems depends on the users' intent and the interpretation of the message by the chatbot or voice-based assistants' system. The following example is a simple conversation between a chatbot and the user.

User: I am looking for a research article.

Bot: Please tell me about the field of study.

User: chemistry

Bot: Please open the link: https://libguides.uta.edu/CHEMInfo

For the thesis and the research, a chatbot is developed using a machine learning-based conversation AI framework: Rasa [19]. I will discuss briefly the architecture of Rasa in high level.

2.3. Rasa Architecture

Rasa consists of two modules: Rasa NLU and Rasa Core.

2.3.1. Rasa NLU:

As the name suggests, Rasa NLU helps the chatbot to understand the users' message (natural language) and extract meaningful information from the users' message. The chatbot system needs to understand the intent of users' messages for the chatbot to reply correctly. The training data will be provided for Rasa NLU to understand the user's intent.

2.3.2. Rasa Core:

Rasa Core handles the flow of conversation between an user and the chatbot. The higher architecture of the Rasa Core [2] is:

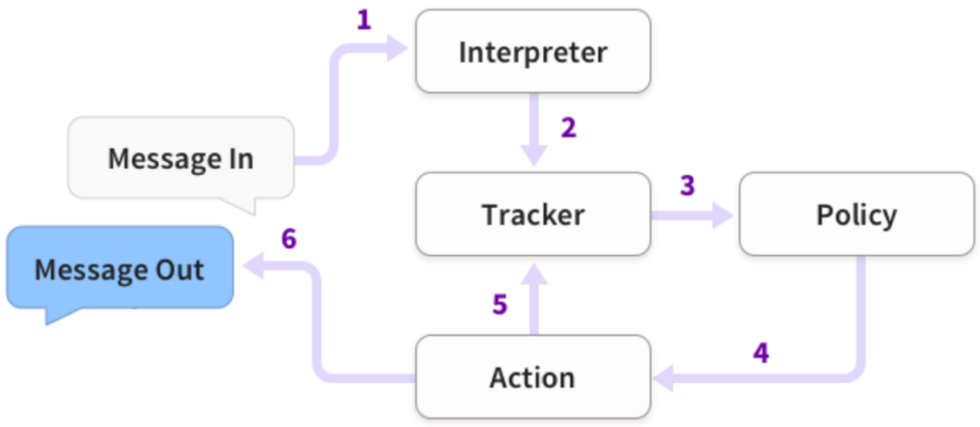

Fig: 2.3: Rasa Architecture [2]

Since Rasa is an open-source framework, a rasa project can be created once the rasa library is installed in the conda or virtual environment.

2.4. Starting a Rasa Project

Once you successfully install the rasa library in the conda or virtual environment, you can create a rasa project with the command **rasa init --no-prompt** [19]. By default, the command will create the following files:

- __init__.py
- __pycache__
- actions.py
- config.yml
- credentials.yml
- data
    - nlu.md
    - stories.md
- domain.yml
- endpoints.yml
- models

The details of each of the components listed above in the official documentation of Rasa or the official GitHub repository. as rasa is an open-source project. I will only show how to include training data in the nlu.md which is present inside the directory data.

intent:greet

- hey

- hello

- hi

- good morning

- good evening

- hey there

- Hello

- Hi

- HI

intent:affirm

- yes

- indeed

- of course

- that sounds good

- correct

- yeah

- Yeah

- yep

The above file only consists of only two intents and few training examples. The name of the intent is defined by the designer of the project. In the above example, some of the possible training data is provided for each of the intent. To increase the domain knowledge for each of the intent, it is desirable to provide as many significant training data as possible. Furthermore, the training dataset for the given intents does not include misspelling although some of the similar words are included. To handle the case where a user misspells a word while communicating with the chatbot, the possible misspellings should also be included in the training dataset.

After providing different paths in stories.md inside the directory data, and other requirements of the rasa project, the model can be trained. In the trained model, a user can communicate with his message, and Rasa NLU will classify the user's utterance with the score between 0 and 1. In the model, we developed, if a user asks: "Where is Engineering Research Building?", the model will classify the user's message to the intent with the highest confidence score, and gives response according to the respective intent. The table below shows the confidence score for the above user's input message.

Intents	Confidence Score
quantitative_research_article	0.67
greet	0.07
choose	0.05
guest_visitor_alumni_wireless_internet	0.03
nursing_librarians_email	0.02
affirm	0.02
article_is_peer_reviewed	0.01
goodbye	0.01
deny	0.0

Table 2.4: Confidence score for an utterance

As the confidence score for the intent "quantitative_research_article" is highest, the user's message will be classified with the intent "quantitative_research_article." As per the path defined in the stories.md and response message, the chatbot will respond with the corresponding action or the response. The issue arises when misspellings or similar words are not included in the training data. The following figure shows one of the scenarios where a user misspells the word "hi" with "hy" but the chatbot is not able to handle it accordingly.

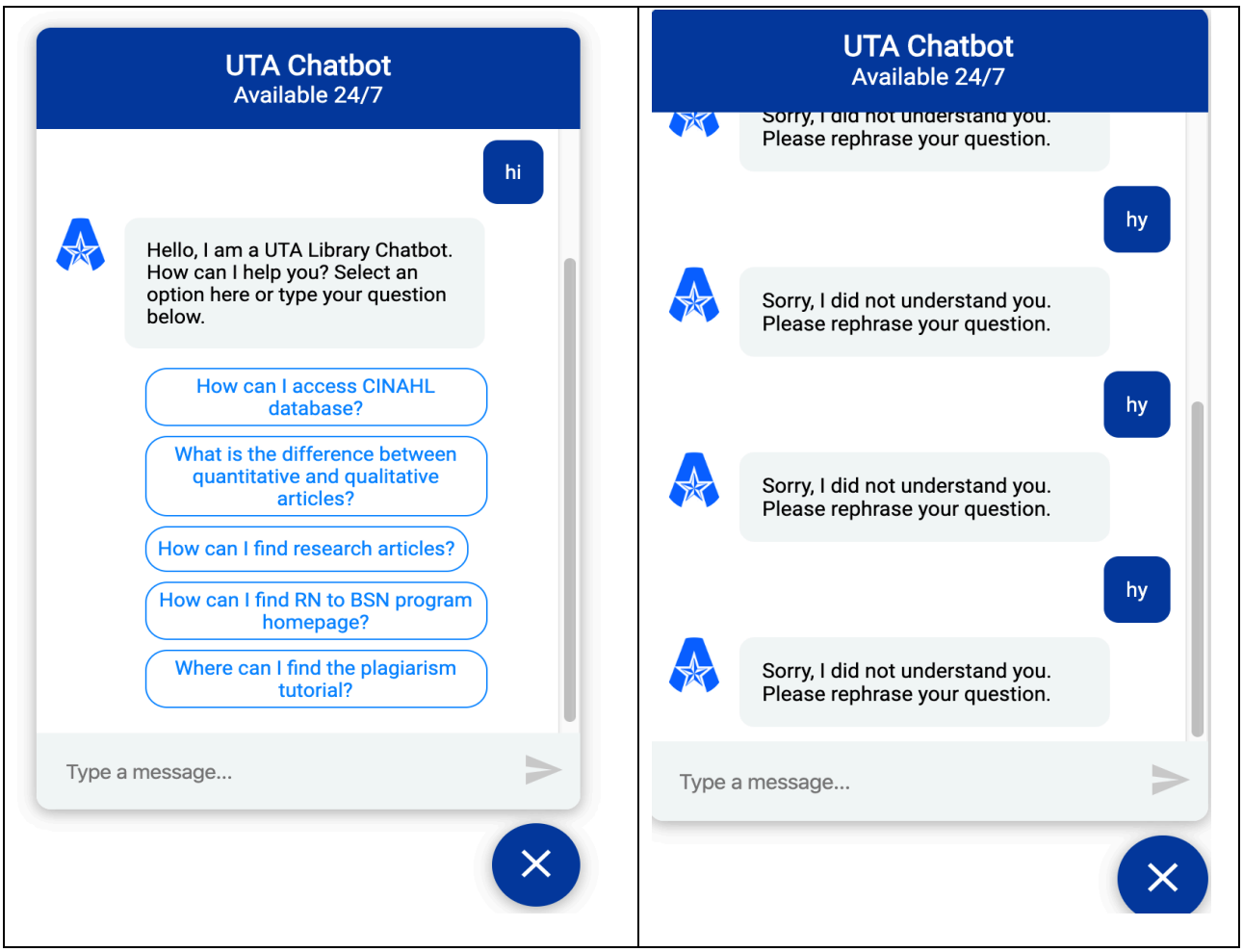

Fig 2.4: Chatbot Conversation without misspelling and with misspelling error [21]

Thus, in this research, we will try to handle scenarios where a user will misspell a word or provide similar words for the given intents. For instance, here in the above example, the Rasa model has not seen the word "hy" while training it, so the model is not sure how to handle the user message "hy." If the word "hy" is included in the training data for the intent, the confidence score of "hy" will be higher than the current score for the intent greet. Thus, to get misspellings, initially, the brute-force approach to generate misspellings is used with an edit distance of 1.

CHAPTER 3

SPELLING GENERATION USING EDIT DISTANCE

3.1. Levenshtein distance

Levenshtein distance [13], commonly referred to as edit distance, is named after the Russian scientist Vladimir I. Levenshtein. The Levenshtein distance is equal to the minimum number of operations required to transform one string to the next string. For instance, the edit distance between the string "tim" and the string "him" is 1. He devised the algorithm in 1965. The common use of edit distance is for speech recognition, spelling checking and correction, plagiarism detection, DNA analysis, and many other practical applications [18]. The operations involved in edit distance is an insertion, deletion, or substitution of the character in the strings. Each operation involved is given a weight of 1. The edit distance is calculated using dynamic programming and the algorithm to find an edit distance between two string M and N is:

Initialization

$D(i, 0) = i$

$D(0, j) = j$

Recurrence Relation:

For each I = 1…M

For each j = 1 …N

$$D(i, j) = \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + \begin{cases} 2 ; & \text{if } X(i) \neq Y(j) \\ 0 ; & \text{if } X(i) = Y(j) \end{cases} \end{cases}$$

Termination:

D(N, M) is an edit distance

Fig 3.1: Algorithm for edit distance

3.2. Generating misspelling using edit distance of 1

One of the ways to generate wrong spelling for any words is to use the edit distance of 1 or greater. Here, I will be generating all the possible words that can be generated with the edit distance of 1.

Furthermore, some of the previous findings and assumptions were included while deciding this approach to use for misspelling generation.

- The first character of the word will not be misspelled while typing.
- Only misspelling of 1 edit distance will be generated.
- Only lower case misspellings will be included.
- 80 % of misspelling happens within the edit distance of 1 [5].

For my research process, 10 words are chosen: hi, bye, chemistry, nursing, accounting, mathematics, psychology, history, fuck, and thanks. These words are chosen based on the fact that "hi" is commonly used for greeting, "bye" and "thanks" are used commonly used to end the communication. Chemistry, nursing, accounting, mathematics, psychology, and history are some of the common subjects or topics whose database link is frequently asked to the research coaches in the library at the University of Texas at Arlington. Furthermore, "fuck" [10] is used to handle any kind of insults to the system. Here, we are not doing any sort of automatic spelling corrections, but only focusing on the generation of misspelling so that the domain knowledge of the chatbot becomes wider to handle common human misspellings.

The characters used for this approach are: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

We will generate misspelling using the following common techniques.

1. **Insertion**: The misspellings will be generated by adding a character one at a time in the chosen word starting from position 1 until the length of the word. The following tables show how insertion is done for the word "thanks."
2. **Deletion**: A character from the given word is deleted one at a time except for the first character.
3. **Substitution:** Each character is substituted with other characters one at a time from the list of lower case English characters.

The table below shows the brute-force method used to generate the wrong spellings for "thanks" and the number of misspellings for insertion, deletion, and substitution with the edit distance of 1. By using this approach, 286 possible misspellings were generated for the word "thanks."

Word	**Insertion**	**Deletion**	**Substitution**
thanks	➢ t_hanks ➢ th_anks ➢ tha_nks ➢ than_ks ➢ thank_s ➢ thanks_ ➢ …	➢ tanks ➢ thnks ➢ thaks ➢ thans ➢ thank	➢ taanks ➢ thbnks ➢ thaaks ➢ thanas ➢ thanka ➢ …
	Total words: 156	Total words: 5	Total words: 125
Total Misspelling:	286		

Table 3.2.1: Generating misspellings for word "thanks"

The following table shows all the possible misspellings that can be generated using the edit distance of 1 with insertion, deletion, and substitution.

Referenced words	Insertion	Deletion	Substitution	Total Misspellings
hi	52	1	25	78
bye	78	2	50	130
chemistry	234	8	200	442
nursing	182	6	150	338
accounting	260	9	225	494
mathematics	286	10	250	546
psychology	260	9	225	494
history	182	6	150	338
thanks	156	5	125	286
fuck	104	3	75	182

Table: 3.2.2: Number of possible misspellings for 10 chosen words.

While using this approach, one of the obvious questions arises: "Do all of these misspellings happen in the real world?" Some of the misspellings generated may not be misspellings at all. Some of the words generated using this brute-force approach will never occur in real conversation. For instance, one of the words that are generated using the word "nursing"

is "nursling." These two words have an edit distance of 1, but "nursling" is a word that is a word in the English vocabulary and has its meaning. Including the words that exist on its own in the English vocabulary will instead result in undesirable effect while training the rasa model, and further introducing the possible errors in the communication and the classification of the word in the Rasa system.

Furthermore, the question of "Does all the wrong spelling exists in everyday conversation" still prevails. One of the possible reasons is the position of keys in the QWERTY keyboard layout. In the paper "A Speech Correction Program Based on a Noisy Channel Model" from AT&T Bell Laboratories [11], the researchers have shown that the number or the count of addition, deletion, insertion, and substitution of one character with the next character is not same. The following two diagrams shows the findings from the paper.

del[X, Y] = Deletion of Y after X

X \ Y (Deleted Letter)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	7	58	21	3	5	18	8	61	0	4	43	5	53	0	9	0	98	28	53	62	1	0	0	2	0
b	2	2	1	0	22	0	0	0	183	0	0	26	0	0	2	0	0	6	17	0	6	1	0	0	0	0
c	37	0	70	0	63	0	0	24	320	0	9	17	0	0	33	0	0	46	6	54	17	0	0	0	1	0
d	12	0	7	25	45	0	10	0	62	1	1	8	4	3	3	0	0	11	1	0	3	2	0	0	6	0
e	80	1	50	74	89	3	1	1	6	0	0	32	9	76	19	9	1	237	223	34	8	2	1	7	1	0
f	4	0	0	0	13	46	0	0	79	0	0	12	0	0	4	0	0	11	0	8	1	0	0	0	1	0
g	25	0	0	2	83	1	37	25	39	0	0	3	0	29	4	0	0	52	7	1	22	0	0	0	1	0
h	15	12	1	3	20	0	0	25	24	0	0	7	1	9	22	0	0	15	1	26	0	0	1	0	1	0
i	26	1	60	26	23	1	9	0	1	0	0	38	14	82	41	7	0	16	71	64	1	1	0	0	1	7
j	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
k	4	0	0	1	15	1	8	1	5	0	1	3	0	17	0	0	0	1	5	0	0	0	1	0	0	0
l	24	0	1	6	48	0	0	0	217	0	0	211	2	0	29	0	0	2	12	7	3	2	0	0	11	0
m	15	10	0	0	33	0	0	1	42	0	0	0	180	7	7	31	0	0	9	0	4	0	0	0	0	0
n	21	0	42	71	68	1	160	0	191	0	0	0	17	144	21	0	0	0	127	87	43	1	1	0	2	0
o	11	4	3	6	8	0	5	0	4	1	0	13	9	70	26	20	0	98	20	13	47	2	5	0	1	0
p	25	0	0	0	22	0	0	12	15	0	0	28	1	0	30	93	0	58	1	18	2	0	0	0	0	0
q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0
r	63	4	12	19	188	0	11	5	132	0	3	33	7	157	21	2	0	277	103	68	0	10	1	0	27	0
s	16	0	27	0	74	1	0	18	231	0	0	2	1	0	30	30	0	4	265	124	21	0	0	0	1	0
t	24	1	2	0	76	1	7	49	427	0	0	31	3	3	11	1	0	203	5	137	14	0	4	0	2	0
u	26	6	9	10	15	0	1	0	28	0	0	39	2	111	1	0	0	129	31	66	0	0	0	0	1	0
v	9	0	0	0	58	0	0	0	31	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	1	0
w	40	0	0	1	11	1	0	11	15	0	0	1	0	2	2	0	0	2	24	0	0	0	0	0	0	0
x	1	0	17	0	3	0	0	1	0	0	0	0	0	0	0	6	0	0	0	5	0	0	0	0	1	0
y	2	1	34	0	2	0	1	0	1	0	0	1	2	1	1	1	0	0	17	1	0	0	1	0	0	0
z	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
@	20	14	41	31	20	20	7	6	20	3	6	22	16	5	5	17	0	28	26	6	2	1	24	0	0	2

add[X, Y] = Insertion of Y after X

X \ Y (Inserted Letter)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	15	1	14	7	10	0	1	1	33	1	4	31	2	39	12	4	3	28	134	7	28	0	1	1	4	1
b	3	11	0	0	7	0	1	0	50	0	0	15	0	1	1	0	0	5	16	0	0	3	0	0	0	0
c	19	0	54	1	13	0	0	18	50	0	3	1	1	1	7	1	0	7	25	7	8	4	0	1	0	0
d	18	0	3	17	14	2	0	0	9	0	0	6	1	9	13	0	0	6	119	0	0	0	0	0	5	0
e	39	2	8	76	147	2	0	1	4	0	3	4	6	27	5	1	0	83	417	6	4	1	10	2	8	0
f	1	0	0	0	2	27	1	0	12	0	0	10	0	0	0	0	0	5	23	0	1	0	0	0	1	0
g	8	0	0	0	5	1	5	12	8	0	0	2	0	1	1	0	1	5	69	2	3	0	1	0	0	0
h	4	1	0	1	24	0	10	18	17	2	0	1	0	1	4	0	0	16	24	22	1	0	5	0	3	0
i	10	3	13	13	25	0	1	1	69	2	1	17	11	33	27	1	0	9	30	29	11	0	0	1	0	1
j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
k	2	4	0	1	9	0	0	1	1	0	1	1	0	0	2	1	0	0	95	0	1	0	0	0	4	0
l	3	1	0	1	38	0	0	0	79	0	2	128	1	0	7	0	0	0	97	7	3	1	0	0	2	0
m	11	1	1	0	17	0	0	1	6	0	1	0	102	44	7	2	0	0	47	1	2	0	1	0	0	0
n	15	5	7	13	52	4	17	0	34	0	1	1	26	99	12	0	0	2	156	53	1	1	0	0	1	0
o	14	1	1	3	7	2	1	0	28	1	0	6	3	13	64	30	0	16	59	4	19	1	0	0	1	1
p	23	0	1	1	10	0	0	20	3	0	0	2	0	0	26	70	0	29	52	9	1	1	1	0	0	0
q	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
r	15	2	1	0	89	1	1	2	64	0	0	5	9	7	10	0	0	132	273	29	7	0	1	0	10	0
s	13	1	7	20	41	0	1	50	101	0	2	2	10	7	3	1	0	1	205	49	7	0	1	0	7	0
t	39	0	0	3	65	1	10	24	59	1	0	6	3	1	23	1	0	54	264	183	11	0	5	0	6	0
u	15	0	3	0	9	0	0	1	24	1	1	3	3	9	1	3	0	49	19	27	26	0	0	2	3	0
v	0	2	0	0	36	0	0	0	10	0	0	1	0	1	0	1	0	0	0	0	1	5	1	0	0	0
w	0	0	0	1	10	0	0	1	1	0	1	1	0	2	0	0	1	1	8	0	2	0	4	0	0	0
x	0	0	18	0	1	0	0	6	1	0	0	0	1	0	3	0	0	0	2	0	0	0	0	1	0	0
y	5	1	2	0	3	0	0	0	2	0	0	1	1	6	0	0	0	1	33	1	13	0	1	0	2	0
z	2	0	0	0	5	1	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4
@	46	8	9	8	26	11	14	3	5	1	17	5	6	2	2	10	0	6	23	2	11	1	2	1	1	2

Fig 3.2.1: Deletion and Insertion of Y after X [11].

sub[X, Y] = Substitution of X (incorrect) for Y (correct)

X \ Y (correct)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0

rev[X, Y] = Reversal of XY

X \ Y	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	2	1	1	0	0	0	19	0	1	14	4	25	10	3	0	27	3	5	31	0	0	0	0	0
b	0	0	0	0	2	0	0	0	0	0	0	1	1	0	2	0	0	0	2	0	0	0	0	0	0	0
c	0	0	0	0	1	0	0	1	85	0	0	15	0	0	13	0	0	0	3	0	7	0	0	0	0	0
d	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0
e	1	0	4	5	0	0	0	0	60	0	0	21	6	16	11	2	0	29	5	0	85	0	0	0	2	0
f	0	0	0	0	0	0	0	0	12	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
g	4	0	0	0	2	0	0	0	0	0	0	1	0	15	0	0	0	3	0	0	3	0	0	0	0	0
h	12	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
i	15	8	31	3	66	1	3	0	0	0	0	9	0	5	11	0	1	13	42	35	0	6	0	0	0	3
j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l	11	0	0	12	20	0	1	0	4	0	0	0	0	0	1	3	0	0	1	1	3	9	0	0	7	0
m	9	0	0	0	20	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	4	0	0	0	0	0
n	15	0	6	2	12	0	8	0	1	0	0	0	3	0	0	0	0	0	6	4	0	0	0	0	0	0
o	5	0	2	0	4	0	0	0	5	0	0	1	0	5	0	1	0	11	1	1	0	0	7	1	0	0
p	17	0	0	0	4	0	0	1	0	0	0	0	0	0	1	0	0	5	3	6	0	0	0	0	0	0
q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	12	0	0	0	24	0	3	0	14	0	2	2	0	7	30	1	0	0	0	2	10	0	0	0	2	0
s	4	0	0	0	9	0	0	5	15	0	0	5	2	0	1	22	0	0	0	1	3	0	0	0	16	0
t	4	0	3	0	4	0	0	21	49	0	0	4	0	0	3	0	0	5	0	0	11	0	2	0	0	0
u	22	0	5	1	1	0	2	0	2	0	0	2	1	0	20	2	0	11	11	2	0	0	0	0	0	0
v	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
w	0	0	0	0	0	0	0	4	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	8	0
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
y	0	1	2	0	0	0	1	0	0	0	0	3	0	0	0	2	0	1	10	0	0	0	0	0	0	0
z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig 3.2.2: Substitution of X for Y and Reversal of XY [11]

The above table indicates that using the brute force method of misspelling generation is not the right approach. For instance, the word "nursling" has an edit distance of 1 with "nursing." But during the experiment, it was found that the character 'l' was added only twice after 's'. Thus, it also validates that weighted Levenshtein distance should be included in the misspellings generation. In the same paper, the researchers have mentioned: "we would hope to extend the prior model to take advantage of context" [11].

For our purpose of generating misspelling, we will be using word embedding to make use of context. "An unsupervised and customizable misspelling generator for mining noisy health-related text sources" published in the Journal of Biomedical Informatics has shown that word embedding can be used for misspelling generation. The initial assumption is that the dataset I will be using for misspelling generation will include misspelling as I selected the dataset from Twitter. Since it is hard to tell how a person can misspell a word exactly, Twitter can be a good source for the misspelling dataset, and I will use word embedding for that purpose. Word embedding [8] embeds semantic information in the word vectors which will be a better approach at finding misspelling than generating a list of words using the method of edit distance of 1.

CHAPTER 4

USING WORD EMBEDDING TO GENERATE MISSPELLINGS AND SIMILAR WORDS

Human beings learn to use different words from different sources. But how can we teach a computer system to use the correct words in context? Does the computer system know the meaning of the words? In most cases, the computer system does not know the meaning of the words. If that is the case, how can you teach a computer system about the next words to use. How can you create a sentence which will have a similar meaning to the words? Does the use of synonyms help to resolve the issue? One of the approaches to address some of the questions is to use the word embedding. To understand word embedding, it is of utmost importance to understand dense vector representation.

4.1. Related Works and Dense vector representation

One of the ways to represent a word in a vector space is to use the concept of the "one-hot" vector. For instance, the word spelling can be represented as [0 1 0 0 0 0]. Here, in this simple example, 1 represents the word spelling and the remaining words of the vocabulary will be 0. As the number of words in a vocabulary increases, the dimension of the one-hot vector will increase as well. Thus, if vocabulary is 20 thousand, then the dimension of the one vector will also be 20 thousand with all 0 but one 1. In such circumstances, it will be nearly impossible to find the similarities between any two words as the value of the cosine similarity will always be close to 0.

In the year 2013, the paper titled "Efficient Estimation of Word Representations in Vector Space" [15] showed the possibility to "compute very accurate high dimensional word vectors from a much larger data set" [15]. Unlike the statistical and rule-based Natural Language Processing where words are treated as atomic units [17], word embedding takes advantage of

words used in the contextual level. After the paper was published, researchers and scholars started to harness the potential of word embedding or word representation in vector space. In a well-trained word embedding model, similar word models will have similar dense vector representation [16]. Furthermore, since misspelling is also semantically similar to a given word, the dense vector representation of a misspelling should also be similar to a given word. In this thesis, I will try to find if the assumption of a dense representation vector of misspellings will be similar to the reference word with an edit distance of 1.

For the research, I have used two pre-trained words embedding model trained on the Twitter dataset: word2vec_twitter_tokens.bin [9] and trig-vectors-phrase.bin [18].

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1.0. Word embedding for Misspelling Generation

The popularity and the use of Twitter have increased significantly in recent days. The tweets of people are not completely free of spelling mistakes and typos. For my research, any well-trained word embedding model trained on the Twitter dataset will return at least some misspellings [17] for the 10 words we chose. In this thesis, I will try to find some of the misspellings within an edit distance of 1 with our reference words. By using this approach, the misspelling found from the word embedding model will be included in the training dataset of the chatbot.

5.1.1. Methodology and Experiment

For the purpose of the misspellings generation, I selected the 500 most similar words for given reference words. From those 500 similar words, the words that are within the edit distance of 1 are selected as the misspelling. The following table shows the number of misspellings within the edit distance of 1 for the given referenced words for the models:

Referenced Words	word2vec_twitter_tokens.bin [9]	#
hi	hii, hai, hie, hy, hei	5
bye	bue, byez, byee, byes, byw, byea, byye, byew, byeb, byr, bbye	11
chemistry	chemistryy, chemisty, chemestry	3
nursing	nurseing, nusing, nursin	3
accounting	accouting, accountin	2
mathematics	mathemathics, mathmatics, mathematic	3
psychology	psycology, phychology	2

history	historyy, histor, hostory, hisyory, historu, hystory, histrory, hstory, histoy, hisory, histry, histoty	12
fuck	furk, fcck, fucvk, fck, fuzk, fucj, fucm, fuckc, fyck, fuvk, fick, fruck, fucq, fhuck, fuk, fhck, fukk, fiuck, fuhk, fucs, furck, fuc, fucjk, fxck, fucc, fuxck, feck, fnck, ffck, fvck, fucx, fuhck, fluck, fuch, fucl, fduck, fauck, fuckl, frck, ffuck, fucck, fuuck, fugk, fuqk, fuwk, fock, fufk, fuckk, fack, fuckz, fuuk, fuxk, fuckx, fyuck	54
thanks	thankes, thanke, thanksz, thanjs, thnks, thanxs, thankd, tthanks, thankx, taanks, thatnks, thankss, thankz, tahanks, thankis, thhanks, thank, thanksh, thans, thaks, thsnks, thamks, thannks, thanku, thancks, thabks, thanka, tganks, thankk, thanksx, tkanks, thaanks, tjanks, thenks, thankq, thaaks, thankks, thanksd, thanls, thanky	40

Table 5.1.1.1: Misspellings with an edit distance of 1 for word embedding word2vec_twitter_tokens.bin [9]

Similarly, the following table shows the misspellings generated using the second model found within the 500 most similar words.

Referenced Words	trig-vectors-phrase.bin [18]	#
hi		0
bye	byee,	1
chemistry	chemisty, chemestry	2
nursing	nurseing, nusing	2
accounting		0
mathematics	mathmatics, mathematic	2
psychology	pschology, psycology	2
history		2
fuck	fucn, fuckk, fuckl, fack, fck, fucks, fuk, feck, fuckn, fuuck, fuc, fuxk	12
thanks	thankx, thsnks, thankd, thankz, thanku, tthanks, thaks, thank, thannks, thatnks, thanky, thanxs, thankls, thans, thamks, thankss, thanls, thnks, thanksa, thabks, thhanks, trhanks, thanka, thaanks	24

Table 5.1.1.2: Misspellings with an edit distance of 1 for trig-vectors-phrase.bin [18]

The number of misspellings for each of the referenced words from this model is significantly less as compared with Table 3.2.2, I can have some misspellings with the combination of edit distance of 1 and word embedding.

5.1.2. Results

By combining the misspelling from the given two models, the table shows all the misspellings combined from both models.

Reference Word	Combining Misspellings from both models	#
hi	hy, hei, hii, hai, hie	5
bye	byee, byye, byew, byez, bbye, byes, byr, bue, byeb, byw, byea	11
chemistry	chemestry, chemisty, chemistryy	3
nursing	nurseing, nusing, nursin	3
accounting	accouting, accountin	2
mathematics	mathmatics, mathematic, mathemathics	3
psychology	psycology, pschology, phychology	3
history	historyy, hystory, histor, hisyory, historu, histoty, histry, hisory, histrory, hstory, histoy, hostory	12
fuck	feck, fucs, fucj, fduck, fuckk, fruck, frck, fuckl, ffck, fucjk, fuch, fuvk, fuxck, fucn, fucl, fuqk, fluck, fucq, furk, fucx, fuckc, fvck, fuk, fack, fucvk, fuc, fhuck, fugk, fyuck, fck, fuckn, fock, fxck, fiuck, fick, fhck, fuhk, fuwk, fucm, fuhck, fnck, fucc, fuckz, fucks, fuzk, fuuk, fucck, fyck, furck, fcck, fuckx, fufk, fuxk, fukk, fauck, ffuck, fuuck	57
thanks	thancks, thaaks, thanjs, tjanks, thanksd, thankss, thankks, thanke, thankls, thanksz, thnks, thamks, thankz, thanku, thatnks, tahanks, thenks, thankq, thannks, thankd, thaks, thanls, thankk, thanksa, thaanks, thanky, thanka, thsnks, thankis, thank, tganks, thabks, taanks, thankx, trhanks, thhanks, thanxs, tkanks, thankes, thans, thanksx, thanksh, tthanks	43

Table 5.1.2: Combined Misspellings from both models

5.2. Word embedding for Similar Words Generation

Since semantically similar words will have similar dense vector representation in the word embedding model, to generate more training data for the selected 10 words, first I will find the 30 most similar words using the cosine similarity. The table below shows the 30 most similar words for the pre-trained model. The first table will show all the 30 most similar words.

5.2.1. Results from word2vec_twitter_tokens.bin [9]

Starting with the first model word2vec_twitter_tokens.bin [9]

Reference word	30 most similar words
hi	Hi, hii, hiii, hiiii, hello, hiya, hiiiiii, hiiiiiii, hiiiii, hey, Hii, heyy, HI, hellooo, Hiii, hiiiiiiii, hellow, Hello, hellooooo, heyyy, heey, hiiiiiiiiiii, helloo, heeeey, haii, hallo, heeeeeeey, hellloo, heeey, hullo
bye	Bye, byee, byeee, byeeee, byeeeee, byeeeeee, byebye, BYE, gbye, Byeee, byyye, Byee, byyyye, Byeeee, byeeeeeee, byeeeeeeee, buhbye, 25ha, Byeeeee, Byebye, byeeeeeeeee, toodles, bbye, bubye, adios, byyyeee, pce, byyee, byE, gdbye
chemistry	biology, physics, Chemistry, chem, geometry, calculus, sociology, math, chemisty, maths, geography, psychology, trig, precal, calc, pre-cal, Biology, 25hanks25ph, microeconomics, biochem, precalc, microbiology, science, geology, oceanography, alg_NUMBER_, ochem, Physics, zoology, economics
nursing	Nursing, cosmetology, doctoral, veterinary, urology, nurse, radiology, primary/secondary, postgraduate, pre-law, work-study, sonography, physc, phlebotomy, midwifery, neurology, neonatal, biomed, bar-tending, optometry, pgce, dental, first-aid, pre-med, premed, oncology, liberal-arts, poli-sci, haircutting, post-secondary
accounting	calculus, physics, Econ, econ, economics, calc, Accounting, biochem, biology, microeconomics, oceanography, acctg, sociology, macroeconomics, pre-calc, chem, geometry, math, maths, geography, pharmacology, precal, anthropology, precalculus, econometrics, ochem, psychology, Calculus, orgo, pre-cal
mathematics	mathematic, physics, maths, math, Mathematics, biology, geography, Maths, calculus, pharmacology, thermodynamics, chemistry, microeconomics, arithmetic, sciene, psychology, science, Physics, physcology, trigonometry,

	accounting, Chemistry, economics, phonology, “maths”, phonetics, biochem, macroeconomics, immunology, geometry
psychology	sociology, biology, psych, geography, anthropology, physics, microbiology, physiology, phycology, economics, criminology, microeconomics, science, oceanography, pysch, hanks, phy, biochem, Sociology, physcology, macroeconomics, linguistics, chemistry, pharmacology, Biology, calculus, anthro, maths, zoology, calc, humanities
history	History, HISTORY, hanks, hist, geography, hanks, #history, history-, chemisty, polisci, chemistry, psychology, economics, histories, sociology, sciene, hanks, theology, histo, science, literature, oceanography, hanks, phy, biology, historyyy, philosophy, macroecon, phyics, PreCalc, macroeconomics
thanks	thx, thanx, Thanks, thnx, thnks, thxs, hanks, thankyou, thanka, thnxs, Thanx, thanks, thabks, thaks, thks, Thx, hanks, thanksss, thanxs, tks, 26hank, thankz, “thanks”, tnx, hanks, thankssss, thankd, thxx, Thankyou, thanku
fuck	fck, fuxk, fxck, fuvk, fvck, f**k, fcuk, fucc, fk, f*ck, “fck”, fuckk, fuk, phuck, f-ck, fugg, eff, fukk, fuq, fucl, fucj, fucck, fxk, fuc, fukc, fucccck, fuhk, fuuck, fawk, fu*k

Table 5.2.1.0: 30 most similar words using cosine similarity using word2vec_twitter_tokens.bin [9]

In the above table, we can observe that some similar words for the words: hi, bye, thanks, and fuck, may be directly included in our training dataset for the respective intents. But for remaining words, the designers of the chatbot who are using the machine learning framework must decide based on the domain knowledge and the prior experience. For the thesis, the chatbot replies to the user with the database of the subject link.

The available subjects are:

1. Accounting	21. Electrical Engineering	41. Military Science
2. Aerospace engineering	22. Engineering	42. Modern Language
3. Anthropology	23. English	43. Music
4. Architecture	24. Environmental and Sustainability Studies	44. Nursing
5. Art + Art History		45. Other
6. Bioengineering	25. Environmental Engineering	46. Philosophy and Humanities
7. Biology		
8. Business	26. Film Studies	47. Physics
9. Career	27. Finance and Real Estate	48. Political Science

10. Chemistry/Biochemistry 11. Citation Help 12. Civil Engineering 13. Communication Studies 14. Computer Science Engineering 15. Criminology & Criminal Justice 16. Data 17. Disability Studies 18. Earth and Environmental Sciences 19. Economics 20. Education	28. Government information 29. History 30. Indus & Manufacturing Systems Engineering 31. Info Systems and Operation Management 32. Interdisciplinary Studies 33. Kinesiology 34. Law 35. Linguistics and TESOL 36. Management 37. Marketing 38. Materials Science Engineering 39. Mathematics 40. Mechanical Engineering	49. Psychology 50. Public Health 51. Public Health Engineering 52. Social Work 53. Sociology 54. Theatre Arts 55. Translational Studies 56. Urban and Public Affairs 57. Videos and Digital Media 58. Women and Gender Studies

Table 5.2.1.1: List of subjects

5.2.2. Training data from word2vec_twitter_tokens.bin [9]

For the research, I only included one word subjects: chemistry, nursing, accounting, mathematics, psychology, kinesiology, and history. But in the table 5.2.1.0, the reference word "chemistry " has similar words like biology, physics, and other subjects that have their database response. So, even though these words are most similar to the word "chemistry," these words will not be included in the training dataset for the intent: chemistry_database. Instead, the words like "chem," and "biochem" will only be included. The choice of the word at this stage has to be done manually, but similar words via word embedding gave us more flexibility and confidence to include other semantic similar words in the training dataset. The following table shows some of the supervision and choices I made to include other words to be included in the training dataset.

Reference word	Selected similar words from 30 most similar words
hi	Hi, hii, hiii, hiiii, hello, hiya, hiiiiii, hiiiiiii, hiiiii, hey, Hii, heyy, HI, hellooo, Hiii, hiiiiiiii, hellow, Hello, hellooooo, heyyy, heey, hiiiiiiiiiii, helloo, heeeey, haii, hallo, heeeeeeey, hellloo, heeey, hullo
bye	Bye, byee, byeee, byeeee, byeeeee, byeeeeee, byebye, BYE, gbye, Byeee, byyye, Byee, byyyye, Byeeee, byeeeeeee, byeeeeeeee, buhbye, byye, Byeeeee, Byebye, byeeeeeeeee, toodles, bbye, bubye, adios, byyyeee, pce, byyee, byE, gdbye
chemistry	chem, biochem, ochem
nursing	Nursing, cosmetology, urology, nurse, radiology, sonography, phlebotomy, midwifery, neurology, neonatal, biomed, optometry, dental, first-aid, pre-med, premed, oncology,
accounting	Accounting
mathematics	mathematic, maths, math, Mathematics, Maths, calculus, arithmetic, , trigonometry, geometry
psychology	psych, phycology, pysch, pyschology, physcology,
history	History, HISTORY, histroy, hist, histor, #history, history, histories, hostory, theology, histo, **historyyy**
thanks	thx, thanx, Thanks, thnx, thnks, thxs, thankss, thankyou, thanka, thnxs, Thanx, thansk, thabks, thaks, thks, Thx, thannks, thanksss, thanxs, tks, thanxx, thankz, "thanks", tnx, thnaks, thankssss, thankd, thxx, Thankyou, thanku
fuck	fck, fuxk, fxck, fuvk, fvck, f**k, fcuk, fucc, fk, f*ck, "fck", fuckk, fuk, phuck, f-ck, fugg, eff, fukk, fuq, fucl, fucj, fucck, fxk, fuc, fukc, fucccck, fuhk, fuuck, fawk, fu*k

Table 5.2.2 Similar words selected from word2vec_twitter_tokens.bin [9]

5.3.1. Results from trig-vectors-phrase.bin [18]

The result obtained by using the other model: trig-vectors-phrase.bin [18]. The following table shows all the 30 most similar words:

Reference word	Trig-vectors-phrase.bin [18]
hi	hello, hey, hiya, howdy, welcome, heya, ds_nuthut, mommahnina, welcome_aboard, deb2210, hugs, smart_smokefree_friends, thanks, smart_smokefree, emmalea, muffindog, congrats, joslene, joined_daily_strength, vasandra, mom2go, emidora, stlbrian, maggiesgirl, welcom, melissa, sherryly, heather, goodmorning, sambod
bye	riddance, goodbye, goodnight, ol_chuggs, byes, bye_bye, riddens, luck, morning_fellow_quitters, postx, morning_teag, copsy, gammie, morning_teegee, ridance, riddence, eenanny, ol_fontucky, gliderkate, ttyl, buh_bye, jazzer, ol, tonyia, tourguide, hunnie_xx, goodbyes, hunny, xoxo_di, omc1
chemistry	biochemistry, biology, chemisty, chemestry, physics, organic_chemistry, maths, biochem, geology, chemistry_biology, chem, physiology, biology_chemistry, microbiology, organic_chem, psychology, computer_science, mathematics, electrical_engineering, physics_astronomy, anthropology, calculus, math, chemistry_organic_chemistry, engineering, sciences, environmental_science, mechanical_engineering, michael_naughton_chairman, marine_biology
nursing	graduate, grad_school, bsn, practicum, bachelors, teaching_assistant, teaching_credential, enrolled, masters_degree, graduating, undergrad, clinicals, vocational_school, crna, doctoral_program, vocational, ashford_university, dental_assisting, college, graduates, clinical_rotations, veterinary_medicine, bachelors_degree, associates_degree, teaching_certificate, elementary_education, bme, mechanical_engineering, ala_accredited, lpn
accounting	finance, finance_accounting, accountancy, actuarial_science, engineering, corporate_finance, software_engineering, accounting_finance, audit, actuarial, supply_chain_management, computer_science, econ, undergraduate, investment_banking, undergraduate_degree, mba, mechanical_engineering, auditing, industrial_engineering, undergrad, poli_sci, electrical_engineering, comp_sci, bookkeeping, environmental_engineering, eecs, chemical_engineering, abet_accredited, civil_engineering
mathematics	mathematical, maths, computer_science, natural_sciences, calculus, physics, applied_mathematics, quantum_mechanics, physics_chemistry_biology, science, social_sciences, math, astrophysics, linguistics, theoretical_physics, biology_chemistry, biology, econometrics, combinatorics, cosmology_astronomy, abstract_algebra, differential_equations, sociology, cosmology, discrete_mathematics, numerical_analysis, particle_physics, quantum_physics, mathematic, sciences
psychology	sociology, anthropology, neuroscience, biology, criminology, biochemistry, social_sciences, humanities, computer_science, natural_sciences, psychology_sociology, sciences, linguistics, sociology_psychology, zoology,

	biology_chemistry, science, astrophysics, philosophy, clinical_psychology_counseling, environmental_science, mathematics, pharmacology, physics_astronomy, behavioral_psychology, poli_sci, applied_mathematics, neurobiology, anthropology_sociology, developmental_psychology
history	histories, ajilsingh, thekingoffunk, theworktop, hitherto_existing_society, puzomezzo, foxaperture, chapar, dan_carlins_hardcore, mparramon, literature, histroy, anthropology, vijayan37, geography, episodic_amnesia, sociology_psychology, marxist_ideology_involving, ithrowpoooo, recent, psychology, naval_architecture_navigation, archaeology, rikijordan73, specific_cave_paintings, psychology_anthropology, annals, thisnews, historian, sociology
thanks	
fuck	f_ck, f_k, feck, fk, fck, fuck_fuck_fuckity, eff, yelly_mcyellerson, fucks, twatfuck, fuuuuck, douchebag_cocksucker_motherfuckers, bitch_cunt_fuckhole, fuuuuuuck, frick, shit, goddamnit, fuckin, god_damnit, fucking, fuuuck, fuc, fuck_fuckity_fuck, fuckity_fuck_fuck, fuuuuuck, fu_k, fuck_fuckety_fuck, fucken, cock_sucker_mother_fucker, fuckitty_fuck]

Table 5.3.1 30 most similar words using cosine similarity using trig-vectors-phrase.bin [18]

From the above table, based on the domain knowledge, the words can be manually selected for the each of the intent.

5.3.2. Training data from trig-vectors-phrase.bin [18]

The chosen words among the 30 most similar words are:

hi	hello, hey, hiya, howdy, hugs, goodmorning
bye	goodbye, goodnight, byes, bye_bye, ttyl, xoxo
chemistry	biochemistry, chemisty, chemestry, organic_chemistry, biochem, chem, organic_chem, chemistry_organic_chemistry,
nursing	bsn, clinicals, crna, dental_assisting, lpn
accounting	finance_accounting, accountancy, actuarial_science, accounting_finance, audit, actuarial, auditing, bookkeeping,
mathematics	mathematical, maths, applied_mathematics, math, combinatorics, abstract_algebra, differential_equations, discrete_mathematics, numerical

psychology	psychology_sociology, sociology_psychology, zoology, clinical_psychology_counseling, behavioral_psychology, developmental_psychology
history	histories, histroy, anthropology, marxist_ideology_involving, archaeology, anthropology, historian
thanks	
fuck	f_ck, f_k, feck, fk, fck, fuck_fuck_fuckity, eff, yelly_mcyellerson, fucks, twatfuck, fuuuuck, douchebag_cocksucker_motherfuckers, bitch_cunt_fuckhole, fuuuuuuck, frick, shit, goddamnit, fuckin, god_damnit, fucking, fuuuck, fuc, fuck_fuckity_fuck, fuckity_fuck_fuck, fuuuuuck, fu_k, fuck_fuckety_fuck, fucken, cock_sucker_mother_fucker, fuckitty_fuck]

Table 5.3.2 : Similar words selected from trig-vectors-phrase [18]

The words will be included in the respective intents in nlu.md for the training of chatbot.

CHAPTER 6

CONCLUSION

By looking at the above findings, the misspellings for any single word can be generated by using word embedding. This technique validates that the misspellings happen in real life, and word embedding showed that the dense vector representation of the misspelling is similar to the words chosen for the research. By considering only 500 most similar words, at least some misspellings were found. Furthermore, word embedding also provided us a way to include more semantically similar words in the training data of the chatbot development.

Thus, by using this approach, the number of training data provided for each of the intent we considered increased significantly, thus increasing the domain knowledge and knowledge base of the chatbot preventing the chatbot to perform poorly or result in undesired and unexpected response [4].

CHAPTER 7

FUTURE WORK

The future work will move beyond a single word. I will see how I can have a similar phrase for words like "nursing student," or "RN-BSN." Furthermore, future work will consider the fact that edit distance will not be limited to 1 only. In this research, misspellings with an edit distance of 2 or higher are not included while finding misspellings. Thus, future research will also include the misspellings with an edit distance of 2 or higher for training purposes. Furthermore, the word embedding only resulted in a single one-word similar words, but the same word may be represented by a phrase. The future work will include training data for the phrase as well as paraphrasing [14][15].

Reference

[1] Conversational AI: The Next Generation Computing Interface. Amazon Alexa. https://developer.amazon.com/en-US/alexa/alexa-skills-kit/conversational-ai (accessed September 22, 2019).

[2] Architecture. Rasa. https://rasa.com/docs/rasa/user-guide/architecture/ (accessed June 11, 2019).

[3] Jason Brenier. An Overview of Conversational AI. GeorgianPartners. https://georgianpartners.com/conversational-ai-overview/ (accessed December 01, 2019).

[4] What Is a Chatbot? Oracle. https://www.oracle.com/solutions/chatbots/what-is-a-chatbot/ (accessed November 04, 2019).

[5] Fred J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM,* 7(3):171-176, 1964.

[6] Jacob Eisenstein. *Introduction to Natural Language Processing*. Cambridge, MA : The MIT Press, 2019. [Online] Available: https://mitpress.ublish.com/ereader/9739/?preview#page/1. (accessed: Sep 11, 2019).

[7] ELIZA: a very basic Rogerian psychotherapist chatbot. ELIZA. https://web.njit.edu/~ronkowit/eliza.html (accessed August 08, 2019).

[8] Word Embedding. TensorFlow. https://www.tensorflow.org/tutorials/text/word_embeddings (accessed December 02, 2019).

[9] Frederic Godin. Improving and Interpreting Neural Networks for Word-Level Prediction Tasks in Natural Language Processing. *PhD thesis, PhD Thesis, Ghent University, Belgium,* 2019.

[10] Hongyu Gong, Yuchen Li, Suma Bhat, and Pramod Viswanath. Context-Sensitive Malicious Spelling Error Correction. *In The World Wide Web Conference (WWW '19).* 2019.

[11] Mark D. Kernighan, Kenneth W. Church, and William A. Gale. A Spelling Correction Program Based on a Noisy Channel Mode. *13th International Conference on Computational Linguistics*; 1990.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems,* pages 1097-1105, 2012.

[13] Vladimir I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady.* Vol. 10: 707. 1966.

[14] Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. Decomposable Neural Paraphrase Generation. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019.

[15] Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. Paraphrase Generation with Deep Reinforcement Learning. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* Oct – Nov 2018.

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings - Workshop at International Conference on Learning Representations*; 2013.

[17] Abeed Sarker and Graciela Gonzalez-Hernandez. An unsupervised and customizable misspelling generator for mining noisy health-related text sources. *Journal of Biomedical Informatics,* 88: 98-107, December 2018.

[18] Abeed Sarker and Graciela Gonzalez-Hernandez. A corpus for mining drug-related knowledge from Twitter chatter: Language models and their utilities. *Data in Brief.* Vol. 10:122-131. 2017.

[19] Rasa Tutorial. Rasa. https://rasa.com/docs/rasa/user-guide/rasa-tutorial/ (accessed August 08, 2019).

[20] Understanding Natural Language Understanding. Understanding Natural Language Understanding. https://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf (accessed July 23, 2019).

[21] Rasa WebChat. BotFront. https://github.com/botfront/rasa-webchat (accessed June 05, 2019)

[22] Joseph Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation,* New York, NY, USA: W. H. Freeman & Co., 1976.