

HOAX CATEGORIZATION

By

Brenda Lee Hooi Fern

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfillment of the requirements

For the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

BUSINESS INFORMATION SYSTEMS

Faculty of Information and Communication Technology
(Perak Campus)

JAN 2015

HOAX CATEGORIZATION

By

Brenda Lee Hooi Fern

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfillment of the requirements

For the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

BUSINESS INFORMATION SYSTEMS

Faculty of Information and Communication Technology
(Perak Campus)

JAN 2015

DECLARATION OF ORIGINALITY

I declare that this report entitled “**HOAX CATEGORIZATION**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

Firstly, I would like to offer my sincerest gratitude and appreciation to my supervisor, Dr. Kheng Cheng Wai, who has supported and guided me throughout the entire project. This project would not have completed in time without his dedicated involvement and assistance in every step of the project.

I would also like to give thanks to my fellow course mates and friends for supporting me emotionally especially when going through tough times throughout the project. Their continuous encouragement and presence when I needed someone to talk to is something I treasure very much. The hard work and effort my lecturers who have taught me for the past 3 years have also been invaluable and have helped me understand various concepts and knowledge that I was not exposed to in the past, and thus also opened up my mind to many new technologies and methods to solve problems. Thank you for all the efforts put in into teaching us.

Lastly, I would like to thank God for keeping me in good health throughout the entire duration of the project and my family for giving me an opportunity to pursue my tertiary education in UTAR. My parents have been my pillar of support whenever I wanted to give up, and have continuously prayed for my success. I am very grateful and thankful for all the sacrifices they have made on my behalf.

ABSTRACT

Categorization and determination of hoaxes have always been an issue, and moreso after the Internet has become part of our lives with the introduction of social networking sites and e-communication. In an attempt to solve this problem, this project aims to produce a Google Chrome extension and a standalone Java application to detect health related hoaxes by extracting the highlighted text from the web page and sends it to the server to query the database to get the top 3 similar links for the user to further read on. The application will also help to categorize if the sentence is a potential hoax or not. To calculate the semantic similarity between the highlighted sentence and the sentences stored in the database, WordNet is used as the English lexical database while using Path, a similarity measure that measure the relatedness of a pair of words based on their path length. A word can have multiple senses, such as the word “fly” that could mean an action that is performed by birds and airplanes, and it could also mean the insect. Therefore, Part-of-Sense (POS) tagging is done on both the highlighted sentence and the sentences that are stored in the database in order to only compare words that are of the same POS when querying the database. To further increase the reliability of the application, synonyms that are in the same synset as the word are also stored in the database so that the sentences queried from the database are not only limited to the same words in the sentence, but also to similar words to the words in the highlighted sentence. Preprocessing is done on the sentences queried, which includes lemmatization to only include meaningful words to obtain more reliable similarity score. Other similarity measures have been reviewed, and this includes Wu & Palmer, Leacock & Chodorow, Li, Resnik, Lin and Jiang measures. Previous works that uses statistical similarity measures such as Cosine and Word Order Similarity as well as on sentence similarity are also reviewed for further understanding and comparison. The application is expected to obtain a precision and recall rate of at least 80%.

TABLE OF CONTENTS

TITLE	i
DECLARATION OF ORIGINALITY	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xi
LIST OF ABBREVIATIONS	xii
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement	1
1.2 Background and Motivation	2
1.2.1 Impact, Significance and Contribution	4
1.3 Project Objectives	5
1.4 Proposed Approach/Study	6
1.4.1 Client-Server Architecture	7
1.4.2 Google Chrome Extension	8
1.4.3 Natural Language Processing (NLP)	9
1.4.3.1 Part of Speech (POS) Tagging	9
1.4.3.2 Lemmatization	10
1.4.4 WordNet	11
1.4.4.1 Path	14
1.4.5 Bipartite Mapping	15
1.4.6 N-grams	15
1.5 Achievement Highlights	16
1.6 Report Organization	16

CHAPTER 2 LITERATURE REVIEW	18
2.1 Literature Review	18
2.1.1 Shortest Path	18
2.1.2 Leacock & Chodorow (<i>lch</i>)	18
2.1.3 Wu & Palmer (<i>wup</i>)	19
2.1.4 Hirst & St-Onge (<i>hso</i>)	19
2.1.5 Resnik (<i>res</i>)	20
2.1.6 Lin et al.	20
2.1.7 Jiang & Conrath	21
2.1.8 Extended Lesk	22
2.2 Review and Comparison of Previous Works	24
CHAPTER 3 SYSTEM DESIGN	28
3.1 Entity Relationship Diagram	28
3.2 Use Case Diagram	29
3.3 Activity Diagram	30
CHAPTER 4 METHODOLOGY, TOOLS AND SYSTEM REQUIREMENTS	34
4.1 Methodology	34
4.2 Tools Used	36
4.2.1 Programming Languages	36
4.2.1.1 Java	36
4.2.1.2 Javascript	37
4.2.1.3 Hypertext Markup Language (HTML)	37
4.2.2 Crawling, Preprocessing and Ranking of Similar Links	37
4.2.3 Development of extension - Google Chrome extension	37
4.2.4 Server	38
4.2.5 Lexical Database	38
4.2.6 Database	38
4.3 System Requirements	39

CHAPTER 5 SPECIFICATIONS, IMPLEMENTATION AND TESTING	40
5.1 System Performance Definition	40
5.2 User Interface Design	42
5.2.1 Standalone Java Application	42
5.2.2 Google Chrome Extension	49
5.3 Verification Plan	52
5.4 Testing Results	53
CHAPTER 6 CONCLUSION	59
6.1 Project Review and Discussions	59
6.2 Project Constraints	60
6.3 Problems Encountered	60
6.4 Future Work and Enhancement	60
BIBLIOGRAPHY	62
APPENDIX A: LIST OF POS TAGS USED IN THE PENN TREBANK PROJECT	A-1
APPENDIX B: SELECTION OF THE N VALUE FOR N-GRAM	B-1
APPENDIX C: RESULTS FROM EXPERIMENTING WITH DIFFERENT VALUES OF N	C-1
APPENDIX D: PRECISION AND RECALL VALIDATION RESULTS	D-1

LIST OF FIGURES

Figure Number	Title	Page
Figure 1-1	A “hoax detection” method	3
Figure 1-2	A detailed analysis on the hoax at hoax-slayer.com	3
Figure 1-3	An example of Facebook users expressing fear over the hoax	4
Figure 1-4	Flow Chart for the Preprocessing Step	6
Figure 1-5	Flow Chart for the Hoax Categorization System	6
Figure 1-6	Basic Client/Server Architecture (Mozilla Developer Network 2015)	7
Figure 1-7	An example of the Adblock extension icon in Chrome	7
Figure 1-8	Architecture for a Chrome extension (Tsonev 2013)	8
Figure 1-9	Parts of Speech in English and Examples	9
Figure 1-10	List of semantic relations ins WordNet and their examples (Miller 1995)	12
Figure 1-11	An Example of a “is-a” Relation in WordNet (Meng, Huang and Gu 2013)	13
Figure 1-12	Examples of a Complete Bipartite Graph (Weisstein, n.d.)	15
Figure 2-1	A Fragment of the WordNet Hierarchy that shows the probability $p(c)$ attached to each content (Greenbacker, n.d.; Lin 1998)	21
Figure 2-2	Overall Similarity between 2 questions	24
Figure 3-1	Entity Relationship Diagram in the Database	28
Figure 3-2	Use Case Diagram	29
Figure 3-3	Activity Diagram for Google Chrome Extension	30
Figure 3-4	Activity Diagram for Hoax Categorization in the Standalone Java Application	31
Figure 3-5	Activity Diagram for Crawling Webpage and Selecting Sentences	32
Figure 3-6	Activity Diagram for Saving Link and Sentence Only	33
Figure 4-1	Rapid Application Development Methodology (Javatechig Resources for Developers 2012)	34
Figure 4-2	System Requirements for Google Chrome Browser	39

(Support.google.com, n.d.)

Figure 5-1	Confusion Matrix for Tabulation of Two-Class Classification Results and the Various Performance Metrics that can be Calculated (Chuah 2014)	40
Figure 5-2	Tab for Verifying a Sentence in the Standalone Application	42
Figure 5-3	Categorizing a sentence in the Standalone Application	43
Figure 5-4	Popup to inform user that there was no sentence entered	43
Figure 5-5	Displaying Categorization Results in the Standalone Application	44
Figure 5-6	Screen For Adding New Sentences	45
Figure 5-7	Popup informing the user that the link exists in the database	45
Figure 5-8	Popup informing the user that no URL was entered	46
Figure 5-9	Popup informing the user that the URL entered is not valid	46
Figure 5-10	Screen when crawling the webpage	46
Figure 5-11	Inform user that sentence cannot be saved without the link	47
Figure 5-12	Screen after crawling is successful	47
Figure 5-13	Successfully saved sentence popup message	48
Figure 5-14	Popup to state that the sentence has been successfully saved with existing link	48
Figure 5-15	Screen if there are no similar records in the database	49
Figure 5-16	The Browser Action icon of the Google Chrome Extension	49
Figure 5-17	Screen if there's no sentence highlighted	50
Figure 5-18	The extension has sent sentence to database and awaiting response	50
Figure 5-19	The Related Links to the Highlighted Sentence	51
Figure 5-20	Screen when there are no related links in the database	51
Figure 5-21	Black Box Testing (Softwaretestingfundamentals.com 2010)	52
Figure 5-22	Graph of the Number of Sentences against the Number of Words in the Sentence	53
Figure 5-23	Graph of the Number of Intersected Sentences against n for Sentence 1	54
Figure 5-24	Graph of the Number of Intersected Sentences against n for Sentence 2	55
Figure 5-25	Graph of the Number of Intersected Sentences against n for	55

	Sentence 3	
Figure 5-26	Graph of the Number of Intersected Sentences against n for Sentence 4	56
Figure 5-21	Graph of the Number of Intersected Sentences against n for Sentence 5	56

LIST OF TABLES

Table Number	Title	Page
Table 2-1	Comparison of Different Semantic Similarity Measures (Meng, Huang and Gu 2013)	23
Table 5-1	Black Box Testing Results for Standalone Java Application	58

LIST OF ABBREVIATIONS

<i>API</i>	Application Programming Interface
<i>CSS</i>	Cascading Style Sheets
<i>DOM</i>	Document Object Model
<i>FAQ</i>	Frequently Asked Question
<i>HTML</i>	Hypertext Markup Language
<i>IDE</i>	Integrated Development Environment
<i>JAWS</i>	Java API for WordNet Searching
<i>JDBC</i>	Java Database Connectivity
<i>JDK</i>	Java Development Kit
<i>MB</i>	Megabytes
<i>NER</i>	Named Entity Recognition
<i>NGD</i>	Normalized Google Distance
<i>NLP</i>	Natural Language Processing
<i>ODBC</i>	Open Database Connectivity
<i>POS</i>	Part Of Speech
<i>RAD</i>	Rapid Application Development
<i>RAM</i>	Random Access Memory
<i>SDK</i>	Software Development Kit
<i>SDLC</i>	Systems Development Life Cycle
<i>TF-IDF</i>	Term Frequency – Inverse Document Frequency
<i>UI</i>	User Interface
<i>URL</i>	Uniform Resource Locator

CHAPTER 1: INTRODUCTION

According to the Oxford Dictionary of Current English for Malaysian Students, the term Hoax is defined as a trick intended to make a person believe something that is untrue and act unnecessarily. Hoaxes are sometimes created based on myths, legends and true stories altered by humans to achieve certain goals such as monetary goals via scams and advertising using these hoaxes. These hoaxes can be found almost everywhere on the Internet, from emails to blogs and webpages, and especially on social networking sites such as Facebook and Twitter.

Some hoaxes are harmless; they are only stories that are untrue, posted to embarrass, humiliate or to make fun of a person. However, there are many hoaxes that ask the reader to answer online surveys and to send warning messages to all his/her contacts to warn about a certain virus, which are called virus hoaxes. There are also hoaxes that encourage the reader to delete certain system files, which can ultimately damage the system. An example of this is the hoax on the jdbgmgr.exe virus and SULFNBK.EXE.

1.1 Problem Statement

The purpose of this project is to help readers of articles to distinguish whether the facts are true or false (a hoax). Many posts/links shared by family and friends via Facebook and emails could cause the reader to be confused, whether could it be true or not, and for some, to act differently than usual, for example a hoax listed in Snopes.com (2013) such as canola oil is dangerous as it is toxic may cause a person to avoid all foods that uses or contains canola oil, which is in fact a healthy oil. Readers may also be misinformed of a certain news such as the missing airplane MH370 has been found in the Bermuda Triangle was spread around Facebook (Snopes.com 2014) , however this news is false as the plane has yet to be found to-date (26th March 2015).

Furthermore, according to Radford's (2014) article in news.discovery.com, a hoax went viral in West Africa which claims that salt water is able to prevent or cure Ebola and therefore causing deaths and sicknesses in the area. The hoax continued to spread everywhere including the Internet and via word-of-mouth that other West African countries were also affected, soon many followed its advice and bathed in hot water and salt and drank salt water as a prevention method. Drinking salt water is unhealthy, even causing the deaths of two people and many more fell ill. Therefore, it can be seen that hoaxes gives a false sense of

security (Radford 2014) and can take lives as people takes any information seriously when a deadly disease such as Ebola still on the rise.

This project will also help readers/users to avoid scams; one way is via sharing and liking pages in Facebook. In recent days, there are many pages in Facebook that promises a large number of free products to giveaway, and all the user/reader has to do is to share and like the page. An example listed in Hoax-Slayer (2014) of this is by a Facebook page by Big W., which is not associated with the Australian departmental store Big W, claiming to giveaway hundreds of electronic items such as the Samsung Galaxy S5 and Dell computers by sharing and liking the page. These pages aim to get a large amount of followers for future scams or to sell in the black market for malicious purposes and usually will direct followers to online surveys that ask for personal information.

With the advancement of technology, hoaxes can be easily spread via email, blogs, and social media. Furthermore, according to www.w3schools.com (2015), statistics have shown that Google Chrome is the most used browser, with 62.5%, followed by Firefox 22.9% and Internet Explorer (IE) with 2.0% in February 2015. This shows that the application produced can be accessed and used by majority of the Internet users, therefore reaching a larger audience and many Internet users will be able to use this functionality within their own browser. Moreover, a standalone version is also provided so that users who do not own the Google Chrome browser can use this functionality as well. However, due to the large amount of information required to detect hoaxes from all aspects, this project will only focus on the scope of health related hoaxes.

1.2 Background and Motivation

Hoaxes can be detected by searching it up using search engines such as Google and Bing to see webpages that discuss on the matter, whether it is a hoax or not. In addition, websites such as Hoax-Slayer, Hoax Busters and Snopes constantly update their database on the latest hoaxes that are spread around the internet, and allows the reader to search based on keywords of the hoax, or the type of hoax it is categorized as. These webpages/websites allow the reader to determine if the article read is a hoax or not, and the detailed explanation on how the hoax came about or the source to prove that the article is not a hoax.



Figure 1-1: A “hoax detection” method

In the above figure, it can be seen that Facebook, a social networking site is used to share hoaxes and at the same time, concerned Internet users will warn their friends and family that a certain article/story is a hoax. Further explanation can be found at hoax detection websites such as hoax-slayer.com and snopes.com as follows:

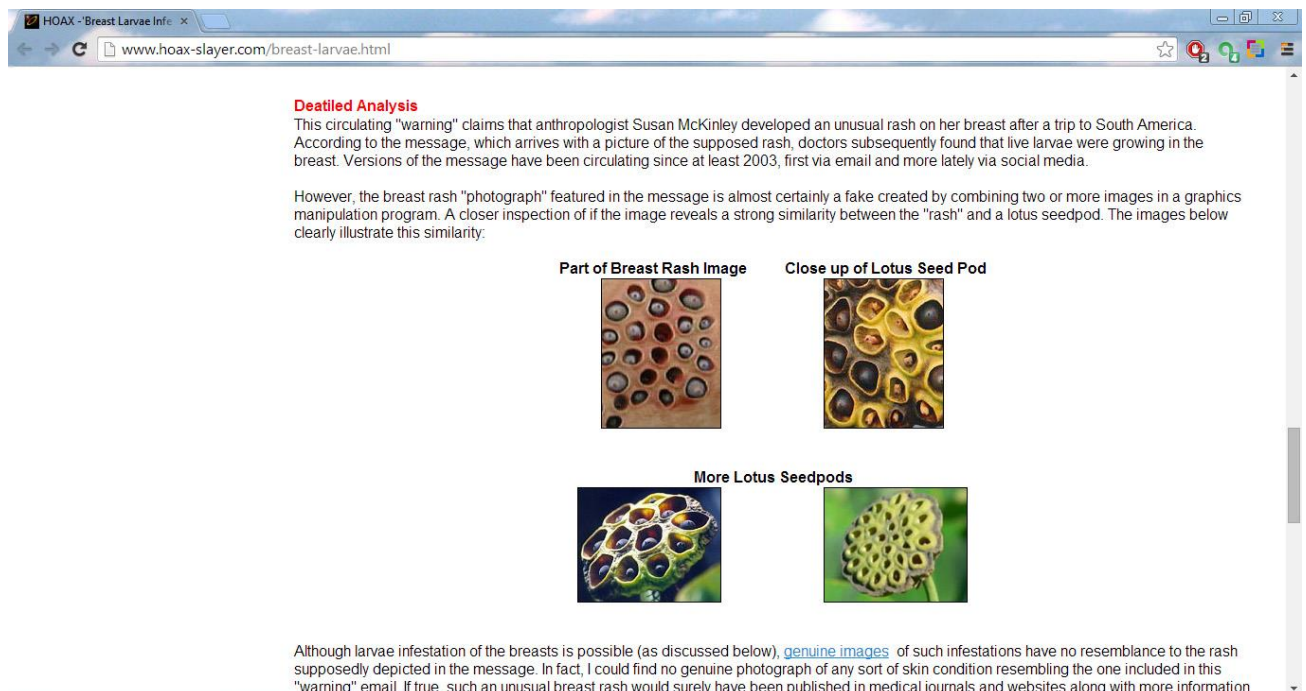


Figure 1-2: A detailed analysis on the hoax at hoax-slayer.com

Even though this is only a hoax, yet it still affects the mental state of readers. Some examples are as follows:



Figure 1-3: An example of Facebook users expressing fear over the hoax

As seen above, it is clear that many are affected by the hoax, and many claim to suffer from the fear of holes. The term tryphobia is frequently used, a term used to describe the fear of holes, although “it's probably not even a real phobia, which the American Psychiatric Association's Diagnostic and Statistical Manual of Mental Disorders says must interfere "significantly with the person's normal routine.”” (Abassi 2011). Thus, this proves that hoax detection is an area that still requires lots of attention to help reduce such anxiety among Internet users.

1.2.1 Impact, Significance and Contribution

The widespread sharing of hoaxes has become increasingly unmanageable, that most people would change their beliefs because of them. Therefore, the contribution by this project is that it will analyze if the sentence against the sentences in the database and categorizes it as a hoax or not, and further provide the links to see related webpages that prove the authenticity

of the message. The focus on health related hoaxes is crucial as these hoaxes can change the lifestyle of a person and therefore the outcome of the project will help to align their beliefs based on facts, and not lies and myths. In addition, it will also ensure that readers/users do not fall into traps and scams created by scammers in an attempt to obtain personal information for malicious purposes.

1.3 Project Objectives

The objectives of this project are as follows:

- 1) To obtain health-related data from reputable websites to store in the database for future retrieval and comparison with queried sentences

Data such as the keywords and a description of the pages from reputable websites such as www.hoax-slayer.com, www.webmd.com and www.snopes.com needs to be extracted and stored in the database so that when a new query comes, it can be compared against the stored data in the database for ranking and categorization.

- 2) To develop a working Google Chrome extension that is able to grab highlighted text and send to server for sentence similarity against sentences in database.

At the end of this project, a Google Chrome extension is expected as the output that is able to extract the highlighted text from a webpage and send this to the database for sentence similarity against the sentences stored in the database. The extension will also display the related links to the highlighted sentence and allow the user to further read more about it in a new tab so as not to disturb their browsing activity.

- 3) To produce a system that has a high precision and recall.

The system should have a precision and recall of 80% to ensure that the chances of selecting the correct link and sentence is higher, thus the results shown to the user would be as accurate as possible so that the facts delivered to the user/reader are only true.

- 4) To find a suitable semantic similarity method to calculate the similarity of sentences and further ranking them in accordance to their similarity to the highlighted sentence.

There are many methods for calculating the similarity between words and sentences. Therefore, many trial and errors have to be done in order to find the most suitable method to be implemented for use in the application.

1.4 Proposed Approach/Study

There are two parts to the project: the preprocessing and the application itself (the analyzing and categorization algorithms are implemented here). The following are the flowcharts for both the preprocessing and implementation stages:

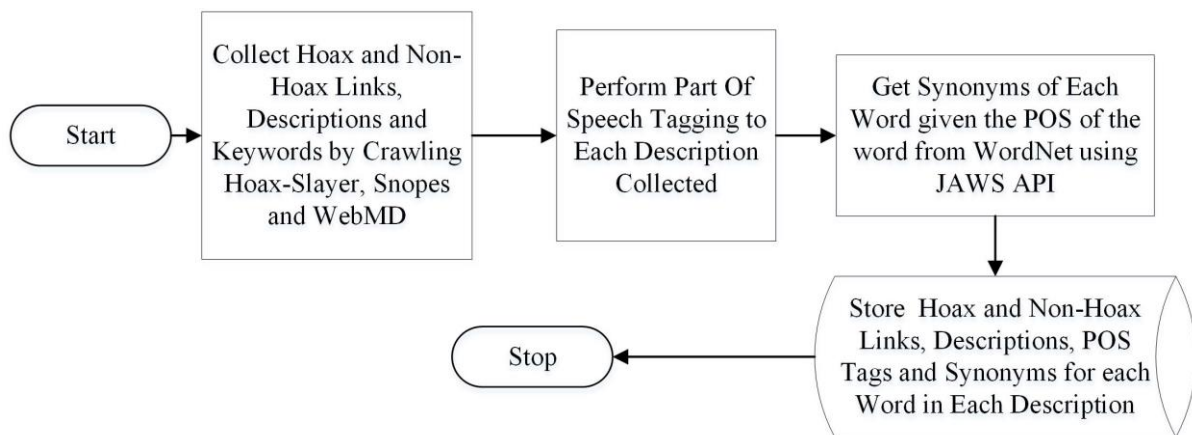


Figure 1-4: Flow Chart for the Preprocessing Step

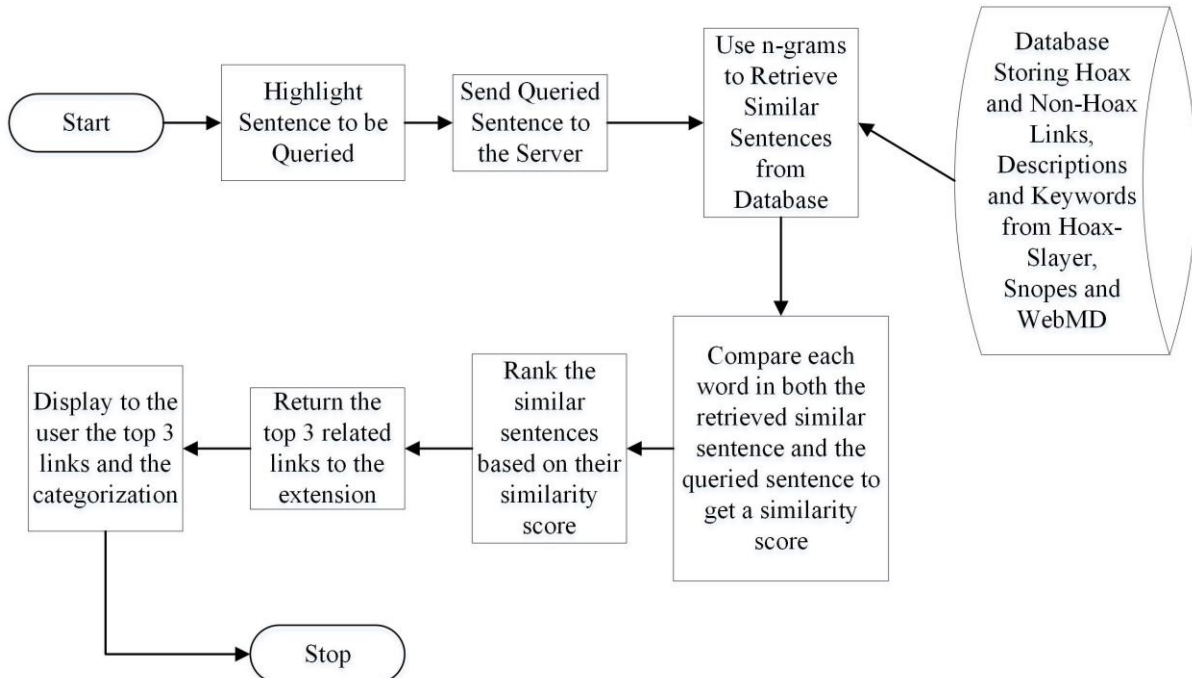


Figure 1-5: Flow Chart for the Hoax Categorization System

1.4.1 Client-Server Architecture

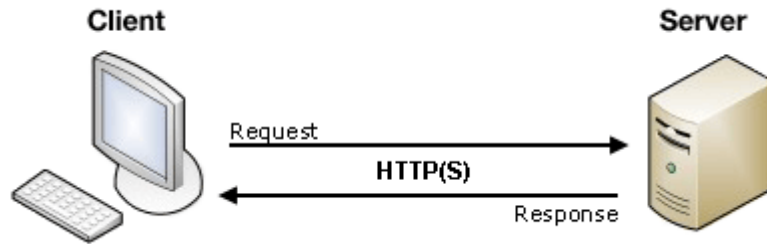


Figure 1-6: Basic Client/Server Architecture (Mozilla Developer Network, 2015)

For this project, the client/server architecture is used for sending data between the client and the server. The client will be the computer that has Google Chrome browser with the extension installed, while the server will receive the request sent by the client, which contains the highlighted sentence, and sends the response, which is the results of the ranking and categorization of the sentence, whether it is a hoax or not.

The reason why this architecture is suitable for this project is because client/server relationship allows more efficient data flow as compared to peer-to-peer networks and allow servers to respond to requests from a large number of clients at the same time (Evans, Martin and Poatsy 2010). The client/server architecture is also centralized, whereby any changes that need to be done to the processing side needs to only be updated in the server side, without affecting the clients. Furthermore, the client/server architecture has increased scalability as compared to other network architecture such as peer-to-peer networks as it allows easy addition of users “without affecting the performance of the other network nodes (computers or peripherals)” (Evans, Martin and Poatsy 2010).

1.4.2 Google Chrome Extension

According to Developer.chrome.com (n.d.), an extension is a small program what modifies and enhances the functionality of the Chrome browser. HTML, JavaScript and CSS are used to write these extension and has little user interface as shown below.



Figure 1-7: An example of the Adblock extension icon in Chrome

Generally, each extension has to have a manifest file that contains information about the extension as well as the allowed permissions/capabilities. HTML, JavaScript and Image files (for icons) are used to display and perform the functionalities that the extension is supposed to do. All these files are packaged into a ZIP file with a .crx suffix (Developer.chrome.com, n.d.) and can be uploaded to the Chrome Extensions Web Store.

Extensions have their own architecture as well, which usually consists of a background page which is further categorized as persistent background pages and event pages, UI pages that interact with the user and content scripts that interact with web pages (Developer.chrome.com, n.d.).

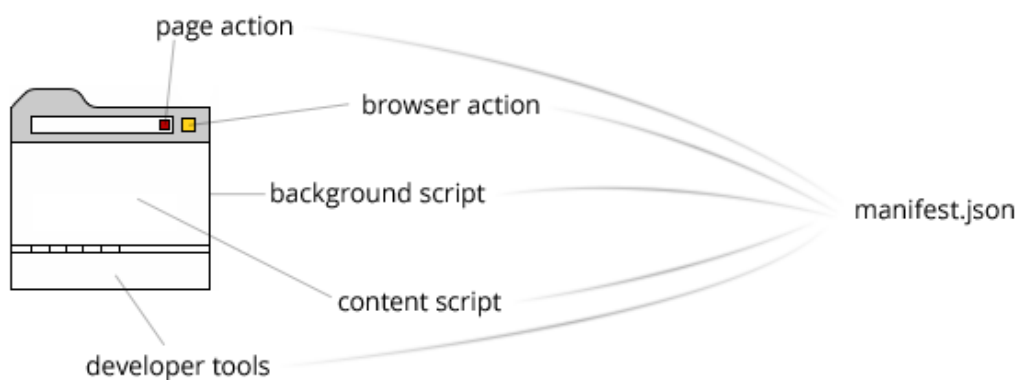


Figure 1-8: Architecture for a Chrome extension (Tsonev 2013)

Background pages can be categorized into persistent background pages where it is constantly running in the background and event pages, which is only called when needed. Event pages are memory saving and helps improve the overall performance of the browser (Tsonev 2013). It is usually used to connect between the other parts of the extension.

For any interaction with the current webpage, the extension would require the content script, which is some JavaScript codes that run on the page that is loaded on the web browser (Developer.chrome.com, n.d.). These scripts allow the developer to read and modify the Document Object Model (DOM) of the webpage, and are possible through passing of message between itself and the extension via Message Passing.

This project utilizes a browser action icon button to interact with the user. This opens up the UI page, which is a popup that shows the current status of the extension and also the top 3 links that are related to the highlighted sentence and the categorization of the highlighted sentence.

1.4.3 Natural Language Processing (NLP)

NLP is to use a computer to analyze natural languages to perform a certain task. It is still an active area of research and is seen to be used in various applications such as robotics, voice recognitions and expert systems. NLP involves various tasks, which includes Part of Speech Tagging, Named Entity Recognition (NER), sentence understanding, machine translation and word sense disambiguation (Nlp.stanford.edu, n.d.). For this project, Stanford CoreNLP and POS Tagger are tools that are used for part-of-speech tagging and lemmatization.

1.4.3.1 Part Of Speech (POS) Tagging

Depraetere and Langford (2012) in their book “Advanced English Grammar: A Linguistic Approach” states that English sentences can be broken down into parts of speech, which are terms to refer to words that behave similarly in sentences. Generally, the parts of speech that are found in sentences are nouns, pronouns, verbs, adjectives, adverbs, prepositions, conjunctions and interjections. However, some authors (such as Depraetere and Langford (2012)) adds the determiner part of speech which according to University of Victoria’s English Language Centre site (Web2.uvcs.uvic.ca, n.d.), determiners include articles such as “the”, “a” and “an”. The following shows some examples of words and their part of speech:

Noun	A noun is a naming word. It names a person, place, thing, idea, living creature, quality, or action. Examples: <i>cowboy, theatre, box, thought, tree, kindness, arrival</i>
Verb	A verb is a word which describes an action (doing something) or a state (being something). Examples: <i>walk, talk, think, believe, live, like, want</i>
Adjective	An adjective is a word that describes a noun. It tells you something about the noun. Examples: <i>big, yellow, thin, amazing, beautiful, quick, important</i>
Adverb	An adverb is a word which usually describes a verb. It tells you how something is done. It may also tell you when or where something happened. Examples: <i>slowly, intelligently, well, yesterday, tomorrow, here, everywhere</i>
Pronoun	A pronoun is used instead of a noun, to avoid repeating the noun. Examples: <i>I, you, he, she, it, we, they</i>
Conjunction	A conjunction joins two words, phrases or sentences together. Examples: <i>but, so, and, because, or</i>
Preposition	A preposition usually comes before a noun, pronoun or noun phrase. It joins the noun to some other part of the sentence. Examples: <i>on, in, by, with, under, through, at</i>
Interjection	An interjection is an unusual kind of word, because it often stands alone. Interjections are words which express emotion or surprise, and they are usually followed by exclamation marks. Examples: <i>Ouch!, Hello!, Hurray!, Oh no!, Ha!</i>
Article	An article is used to introduce a noun. Examples: <i>the, a, an</i>

Figure 1-9: Parts of Speech in English and Examples

Therefore, as a preprocessing step, POS Tagging is done on the sentences to be compared for their similarity using Stanford's POS Tagger. The POS Tagger utilizes a trained tagger model which for this project, the English language tagger model is used. The POS Tagger will take a sentence as its input, and tags each word in the sentence with the appropriate POS. Only words from selected part of speech tags such as adjectives, nouns and verbs are kept in the database as well as taken from the highlighted sentence so that only words deemed as "important" or have contribute to the core meaning of the sentence are considered when calculating the semantic similarity between words in both sentences. Adverbs are not saved as they tend to be words such as "often", "further" and "also", which further illustrates the noun or verb in the sentence but does not carry the main meaning of the sentence. The Stanford POS Tagger utilizes the Penn Treebank tag set which is shown in Appendix A.

1.4.3.2 Lemmatization

To reduce or derive the base form of a word, stemming or lemmatization can be used to achieve this. An example would be to get the base word of "cooking", which is "cook". However, lemmatization is chosen over stemming for this project, which is further explained below.

Stemming uses a crude heuristic process that attempts to obtain the base word of the given word by trying to substitute it with common endings or remove the affixes totally. One of the most used stemming algorithms is the Porter Stemming Algorithm, which is written and maintained by Martin Porter is used to perform this process. However, as it uses a crude method when attempting to obtain the base word, the semantic meaning is no longer taken into account, and the undesired outcome of having stemmed words that have deviated from its original base word may be obtained. An example is "really". After going through the stemming process, it will return the word "realli", which does not carry its original meaning. Therefore, this method is not selected for the project.

On the other hand, the lemmatization process aims to return the dictionary form of the given word, which is known as the "lemma", with "the use of a vocabulary and morphological analysis of words" (Nlp.stanford.edu 2008). Lemmatization takes into account the whole sentence and how the word is being used. For example, the word "saw", of which lemmatization attempts to return "see" or "saw" depending of the POS of the word in the sentence (e.g. if it is a verb or noun) while stemming may return only "s" (Nlp.stanford.edu

2008). Thus, it would seem that lemmatization would help maintain the meaning to the word, and would not affect the semantic similarity score when calculating between words in sentences. The tool used for the lemmatization process is the Stanford CoreNLP by the Stanford Natural Language Processing Group.

1.4.4 WordNet

WordNet is an English lexical database that groups together words (nouns, verbs, adjectives and adverbs) into different concepts which consists of sets of cognitive synonyms called synsets that are linked via conceptual-semantic and lexical relations (Princeton University 2010). Words are evaluated based on their senses which are represented by synonyms that have that sense and are labeled with the semantic relations the word has with other words.

Sense is the meaning of the word in that context, also known as the word sense. For example, the sentences “They went to the park to play” and “The Midsummer Night’s Dream play was very interesting”. Both sentences have the word “play”, however, their meanings differ as in the first sentence the word “play” has the meaning of performing an activity for fun while in the second sentence, the word “play” means a dramatic work that is performed on stage. Since a word can have multiple senses, word sense disambiguation is a part of natural language processing applications.

In WordNet, words are connected from the same part of speech (POS) and therefore consist of four sub-nets: nouns, verbs, adjectives and adverbs. WordNet links words via semantic relations, and according to Miller (1995), there are 6 types of semantic relations in WordNet. The table below shows that WordNet accepts the four POS as mentioned above, therefore, only words that belong in these POS are taken into consideration when performing semantic similarity between words.

Semantic Relation	Syntactic Category	Examples
Synonymy (similar)	N, V, Aj, Av	pipe, tube rise, ascend sad, unhappy rapidly, speedily
Antonymy (opposite)	Aj, Av, (N, V)	wet, dry powerful, powerless friendly, unfriendly rapidly, slowly
Hyponymy (subordinate)	N	sugar maple, maple maple, tree tree, plant
Meronymy (part)	N	brim, hat gin, martini ship, fleet
Troponymy (manner)	V	march, walk whisper, speak
Entailment	V	drive, ride divorce, marry
<i>Note: N = Nouns Aj = Adjectives V = Verbs Av = Adverbs</i>		

Figure 1-10: List of semantic relations in WordNet and their examples (Miller 1995)

Meng, Huang and Gu (2013) explains the relationships in WordNet slightly differently since “language semantics are mostly captured by nouns or noun phrases” and therefore it is the focus of research in semantic similarity calculating. According to their paper, there are four frequently used semantic relations for nouns: hyponym/hypernym (is-a), part meronym/part holonym (part-of), member meronym/member holonym (member-of) and substance meronym/substance holonym (substance-of) (Meng, Huang and Gu 2013). In this structure, it is then shown that the deeper concepts are more specific while the concepts in the upper region are more abstract.

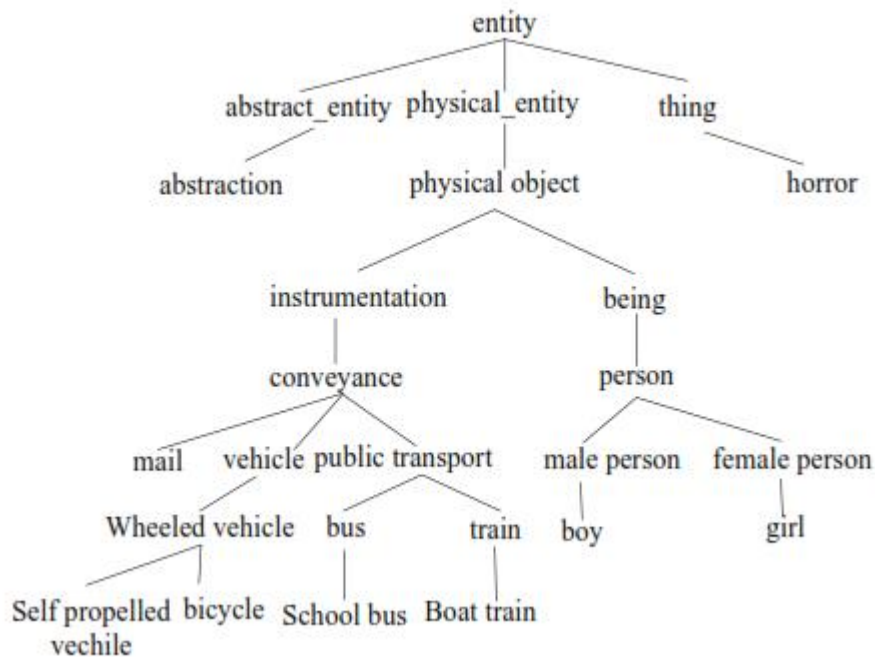


Figure 1-11: An Example of a “is-a” Relation in WordNet (Meng, Huang and Gu 2013)

Synonymy is the main relation among words in WordNet (Princeton University, 2010) and is the symmetric relation between word forms (Miller 1995). This relation relates words that have the same sense. Similarity of the words are evaluated as more similar if the words share more features of meaning (“near-synonyms”) and are less similar if the words have fewer common meaning elements, thus contributing to a greater “semantic distance” (Greenbacker, n.d.).

Antonymy (opposing-name) is the lexical relation between word forms and is also a symmetric semantic relation between word forms (Miller 1995). The antonym of a word “x” is not always “not-x” and therefore semantic relations between word forms and word meanings have to be distinguished clearly (Miller et al. 1993). It forms the principle in the organization of the meanings of adjectives and adverbs. An example would be the “thin” and “fat”. A person who is not thin does not necessarily be fat and vice versa.

Hyponymy (sub-name) or (is-a) relationship accounts for about 80% the relations (Meng, Huang and Gu 2013) and its inverse, hypernymy (super-name) are transitive relations between synsets (Miller 1995). It is the semantic relation between word meanings and since it is normally a single superordinate, a hierarchical semantic structure is formed. It has the parent-child structure, and therefore the hyponym inherits the features the superordinate (parent) and adds at least a feature to distinguish itself from the parent and the other children

hyponyms. For example, a boy “is-a” male, and a girl “is-a” female, and both male and female “is-a” person.

The part-whole (or HASA) relation is known as meronymy (part-name) and its inverse, holonymy (whole-name), and is a complex semantic relations, which in line with Meng, Huang and Gu (2013), can be further categorized as component, substantive and member parts. According to Wordnet’s website by Princeton University (Princeton University, 2010), parts are not inherited “upward” but inherit from their superordinates as there may be certain characteristics that only some things have but not the whole class. For example, the meronymy relation holds synsets like “chair” and “seat” and “leg” however not all furniture have legs even though chairs have legs (Princeton University 2010).

Verbs are structured just like how hyponymy is for nouns, called tryponymy (manner-name) or tryponyms for the verbs with this relation arranges verb synsets into hierarchies, but are much shallower compared to hyponymy. The deeper concepts are more specific manner describing an event such as the volume dimension with verbs like “communicate” – “talk” – “whisper” and the specific manner expressed is depended on the semantic field (Princeton University 2010). Miller (1995) states another relation for verbs called entailment which follows a logic where if a verb X has been done, then verb Y can only be done, and therefore verb X entails verb Y (Wordnet.princeton.edu, n.d.).

1.4.4.1 Path

To calculate the similarity between words in two sentences, ws4j (WordNet Similarity for Java) API is used. Ws4j is the Java version of the WordNet::Similarity Perl implementation from Prof. Ted Pedersen’s group in University of Minnesota in Duluth and is written by Hideki Shima from Carnegie Mellon University (USA) (Shima, n.d.). His API offers eight semantic relatedness metrics, which include Hirst & St-Onge, Jiang & Conrath, Leacock & Chodorow, Wu & Palmer, Lesk, Lin, Resnik and Path. For this project, the path semantic relatedness metric is used as the similarity metric in calculating the semantic similarity.

Path counts the number of nodes along the shortest path between the senses in the ‘is-a’ hierarchies of WordNet to calculate the semantic relatedness of word senses and is inclusive of the end nodes. Therefore, if the two words are in the same concept, the distance between them is one, and thus their relatedness is also one (Pedersen, Patwardhan and Michelizzi, n.d.). This shows that the longer the path length, the relatedness is also lesser.

The relatedness value is the multiplicative inverse of the path length (distance) between the two concepts, and is shown in the equation below:

$$PATH(s1, s2) = \frac{1}{path_{length}(s1, s2)}$$

Where $s1$ and $s2$ are synsets of the two words whose semantic relatedness is to be calculated, and $path_{length}(s1, s2)$ is number of nodes along the shortest path between the senses in the 'is-a' hierarchies of WordNet (Pedersen, Patwardhan and Michelizzi, n.d.).

However, if the two words are not from the same concept/synset, then the value returned will be a large negative number, and for this project, it will be replaced with zero so that it will not affect the overall semantic similarity calculation for both sentences. Thus, path's largest similarity score can only be 1.0 and the minimum score is 0.0. Path will compare between all the senses of both words and select the highest value that is compared.

1.4.5 Bipartite Mapping

To calculate the overall semantic similarity of two sentences, each word in each sentence is treated as a set of vertices and each sentence is a disjoint set as they are initially assumed that there is no element in common. The semantic similarity of each word pair is then the edge between the two vertices from the two disjoint sets as illustrated as follows:

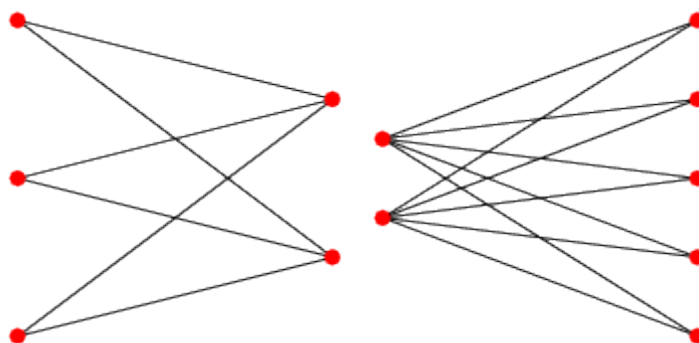


Figure 1-12: Examples of a Complete Bipartite Graph (Weisstein, n.d.)

The outcome for this mapping is a matrix that consists of the semantic similarity between word pairs and from this the highest score between word-pairs are selected for the overall semantic similarity.

1.4.6 N-grams

The N-gram model is used when querying and retrieving sentences from the database. It is illustrated as placing a small window over a sentence that shows n words at a time. When $n = 1$, it is called unigram, and when $n = 2$, it is called bigram and etc. For this project, the highlighted sentence from the user is broken down into n-grams where $n = 5$ and is used for querying the database. For example, a phrase “Curiosity killed the cat but it survived anyway” has 4 n-grams where $n = 5$: {Curiosity, killed, the, cat, but}, {killed, the, cat, but, it}, {the, cat, but, it, survived} and {cat, but, it, survived, anyway}.

Based on the analysis done as shown in the testing results, the number of results obtained when $n = 5$ is not too many or too little. This is also known as a query relaxation method when querying the database for this system. Query relaxation is a method to widen a query so that more records can be retrieved when the original search query returns none or only a few records (Clark 2010). As in the above example, the search algorithm is therefore now $(\text{Curiosity} \cup \text{killed} \cup \text{the} \cup \text{cat} \cup \text{but}) \cap (\text{killed} \cup \text{the} \cup \text{cat} \cup \text{but} \cup \text{it}) \cap (\text{the} \cup \text{cat} \cup \text{but} \cup \text{it} \cup \text{survived}) \cap (\text{cat} \cup \text{but} \cup \text{it} \cup \text{survived} \cup \text{anyway})$.

1.5 Achievement Highlights

At the end of this project, a Google Chrome extension that is able to extract the highlighted text from the webpage and sends it to the server for semantic similarity calculation. The data sent back to the extension includes links to related webpages and informs the user that based on the related links found in the database, the text read is most likely a hoax or not. Furthermore, a standalone Java application is also developed for those who may not own the Google Chrome browser or is unfamiliar with using the Google Chrome extension functionality. The system is able to obtain a precision and recall rate of 80% when comparing with similar sentences from the database. The database contains 59 records from www.hoax-slayer.com, and 259 records from www.snopes.com as records labeled as hoaxes while 395 records from www.webmd.com serves as the non-hoax records.

1.6 Report Organization

The rest of this report consists of Chapter 2 that compares different semantic similarity measures as well as how and what methods previous works have used to solve similar problems. Chapter 3 shows the system’s design and workflow and explains the steps taken to develop the application. Chapter 4 discusses on the methodology and tools used in developing the entire project as well as the system requirements that the user will need to run

CHAPTER 1: INTRODUCTION

the application. Chapter 5 discusses the implementation and testing specifications and results and lastly, Chapter 6 will conclude the entire report by giving a review on the entire project inclusive of the achievements, contributions and objectives achieved as well as some of the issues encountered during the entire project duration and some future improvements that could further enhance the application.

CHAPTER 2: LITERATURE REVIEW

2.1 Literature Review

According to Greenbacker (n.d.), there are two methods to calculate the semantic similarity between words: Thesaurus and Distributional methods. The thesaurus method uses a lexical database such as Wordnet as a thesaurus and measures the distance between two senses while the distributional method estimates the word similarity by finding words that have a similar distribution in a corpus (Greenbacker, n.d.). However, since the database used does not cover all hoaxes and one word can have multiple meanings, therefore the distributional method is not as suitable as using the thesaurus method.

Generally, there are two types of measures: path-based (also known as edge-based or structure-based) and information content (node-based) measures. Further research has brought forward hybrid measures and feature-based (or gloss-based) measures. According to Pedersen, Patwardhan and Michelizzi (2004), there are three similarity measures that are based on path-based, and that includes the Leacock & Chodorow, Wu & Palmer and Path measures. The information content measures include Jiang & Conrath, Resnik and Lin measures.

2.1.1 Shortest Path

Some of the path-based measures depend on the shortest path between the two concepts. The formula is as shown below:

$$Sim(C1, C2) = 2 * Max(C1, C2) - len(C1, C2)$$

Where $Max(C1, C2)$ is the maximum path length between $C1$ and $C2$ and the shortest path relating (minimum number of links) concepts $C1$ and $C2$ (Slimani 2013; Meng, Huang and Gu 2013).

2.1.2 Leacock & Chodorow (*lch*)

The Leacock & Chodorow measure calculates the relatedness similarity of two words by finding the shortest path between two synsets/concepts and further scales the score by the maximum path length in the “is-a” hierarchy (Pedersen, Patwardhan and Michelizzi 2004). The formula is as follows:

$$Sim_{LC}(C1, C2) = -\log\left(\frac{len(C1, C2)}{2 * deep_max}\right)$$

Where $len(C1, C2)$ is the length of the shortest path between two concepts $C1$ and $C2$, and $deep_max$ is the maximum depth of the taxonomy (Slimani 2013; Meng, Huang and Gu 2013).

According to Meng, Huang and Gu (2013), when $C1$ and $C2$ are in the same sense, $len(C1, C2)$ will return 0, and therefore both $len(C1, C2)$ and $2 * deep_max$ needs to add 1 to avoid the situation where $\log(0)$ can occur. Therefore, the ranges of values obtained are between $(0, \log(2 * deep_max + 1)]$.

2.1.3 Wu & Palmer (*wup*)

Wu & Palmer's similarity measure takes the position of the concepts $C1$ and $C2$ to the position of the closest most specific common concept (also known as the lowest common subsumer), $lso(C1, C2)$. The formula is as follows (Meng, Huang and Gu 2013):

$$Sim_{wup}(C1, C2) = \frac{2 * depth(lso(C1, C2))}{len(C1, C2) + 2 * depth(lso(C1, C2))}$$

Where $len(C1, C2)$ is the distance (number of "is-a" links) that separates the concepts $C1$ and $C2$ from the lowest common subsumer $lso(C1, C2)$. $depth(lso(C1, C2))$ is the distance between the root node and the lowest common subsumer for concepts $C1$ and $C2$. The range of values are between $(0, 1]$ (Meng, Huang and Gu 2013).

2.1.4 Hirst & St-Onge (*hso*)

This measure is a path-based measure and classifies relations in WordNet as having direction (Pedersen, Patwardhan and Michelizzi, 2004). Two concepts are semantically close if their synsets are connected to a relatively short path (in Shima's ws4j web demo, the distance is not more than 5) and relatively stationery (does not change direction too often). An example is the "is-a" relation is categorized as upwards, and the "has-part" relation as horizontal. According to Silmani (2013), an Allowable Path is a path that does not stray from "the meaning of the source concept" and therefore is considered when calculating relatedness.

The similarity function is as below (Slimani 2013):

$$Sim_{hso}(C1, C2) = C - SP - k * d$$

Where $C1$ and $C2$ are two concepts in WordNet, d is the number of changes of the direction in the path that connects $C1$ and $C2$ while C and k are constants that are derived from experiments (Slimani 2013).

2.1.5 Resnik (*res*)

The Resnik measure relies on the information content to calculate the word similarity and adds probabilistic information that is derived from the corpus (Greenbacker, n.d.). The measure uses the basis that “two concepts are more similar if they present a more shared information”. The information content of the concepts that subsume the two concepts in WordNet indicates the information shared by the two concepts. It is defined as follows:

$$Sim_{Resnik}(C1, C2) = -\log P(lso(C1, C2)) = IC(lso(C1, C2))$$

Where $C1$ and $C2$ are two concepts in WordNet, and $lso(C1, C2)$ is the lowest common subsumer between the two concepts and $IC(lso(C1, C2))$ is the information content of that subsumes them. P is defined as:

$$P(c) = \frac{\sum_{w \in words(c)} count(w)}{N}$$

Where $words(c)$ is the set of words subsumed by a concept c , and N is the number of words in the corpus and WordNet (Greenbacker, n.d.).

Information such as the size of the corpus is provided by the measure (Slimani 2013). It is also “considered somewhat coarse” because the same least common subsumer is shared with many different pairs of concepts.

2.1.6 Lin et al.

Lin et al. has calculates the similarity based on the hierarchic links and the corpus (Slimani 2013). This similarity measure is based on the more differences between the two concepts, the less similar they are (Greenbacker, n.d.), and is shown as follows:

$$Sim_{Lin}(C1, C2) = \frac{2 * \log P(lso(C1, C2))}{\log P(C1) + \log P(C2)}$$

Where $C1$ and $C2$ are two concepts in WordNet and $lso(C1, C2)$ is the lowest common subsumer for the two concepts. It is based on the similarity theorem where they similarity between A and B is the ratio of the amount of common information of A and B and the information that fully describes A and B (Greenbacker, n.d.).

$$Sim_{Lin}(A, B) = \frac{common(A, B)}{description(A, B)}$$

Therefore, according to Slimani (2013), Lin et al.'s measure gives a better ranking of similarity as compared to Resnik's measure.

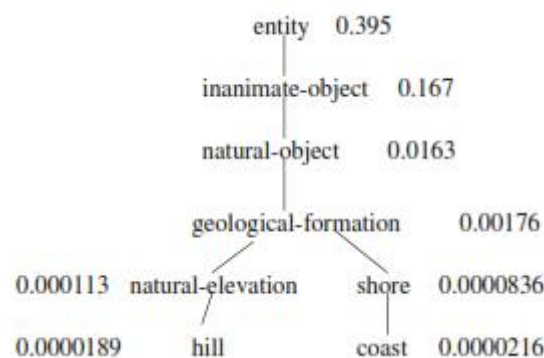


Figure 2-1: A Fragment of the WordNet Hierarchy that shows the probability $p(c)$ attached to each content (Greenbacker, n.d.; Lin 1998)

2.1.7 Jiang & Conrath

Jiang & Conrath's similarity measure calculates the semantic relatedness using a combination of edge counts in the "is-a" hierarchy in WordNet and the information content in values of WordNet concepts. This measure is expressed as distance instead of similarity, and therefore the value is inverted to obtain the semantic relatedness measure. The formula is as below:

$$dist_{JC}(C1, C2) = 2 * \log P(lso(C1, C2)) - (\log P(C1) + \log P(C2))$$

And therefore to obtain the semantic similarity measure:

$$Sim_{JC}(C1, C2) = \frac{1}{2 * \log P(lso(C1, C2)) - (\log P(C1) + \log P(C2))}$$

Where $C1$ and $C2$ are two concepts in WordNet and $lso(C1, C2)$ is the lowest common subsumer for the two concepts.

This measure takes into consideration of the shortest path between the two concepts and the density of the concepts in the same path (Slimani 2013).

2.1.8 Extended Lesk

Lesk was originally proposed by Lesk in 1985, and states that “the relatedness of two words is proportional to the extent of overlaps of their dictionary definitions” (Pedersen, Patwardhan and Michelizzi, n.d.). A gloss, which is a short description that explains the meaning of the concept by the synset, is assigned to each synset. Banerjee and Pedersen (2002) extend and adapt the original Lesk algorithm. Relatedness is calculated by the overlap scores between the glosses of the two concepts and the relationships between concepts in WordNet. Therefore, the extended Lesk measure takes into account not only the glosses, but also the hypernyms, hyponyms, meronyms and other relations (Greenbacker, n.d.). The similarity between two concepts A and B are expressed as follows:

$$\begin{aligned} \text{similarity}(A, B) = & \text{overlap}(\text{gloss}(A), \text{gloss}(B)) \\ & + \text{overlap}(\text{gloss}(\text{hypo}(A)), \text{gloss}(\text{hypo}(B))) \\ & + \text{overlap}(\text{gloss}(A), \text{gloss}(\text{hypo}(B))) \\ & + \text{overlap}(\text{gloss}(\text{hypo}(A)), \text{gloss}(B)) \end{aligned}$$

Which can be expressed in the following formula:

$$\text{sim}_{e\text{Lesk}}(C1, C2) = \sum_{r, q \in \text{RELS}} \text{overlap}(\text{gloss}(r(C1)), \text{gloss}(q(C2)))$$

Where $C1$ and $C2$ are two concepts in WordNet and r and q are relations such as hypernyms, hyponyms, etc.

CHAPTER 2: LITERATURE REVIEW

The following is a table that compares the different semantic similarity measures that is compared and evaluated by Meng, Huang and Gu (2013):

category	Principle	measure	features	advantages	disadvantages
Path based	function of path length linking the concepts and the position of the concepts in the taxonomy	Shortest path	count of edges between concepts	simple	two pairs with equal lengths of shortest path will have the same similarity
		W&P	path length to subsumer, scaled by subsumer path to root	simple	two pairs with the same lso and equal lengths of shortest path will have the same similarity
		L&C	count of edges between and log smoothing	simple	two pairs with equal lengths of shortest path will have the same similarity
		Li	non-linear function of the shortest path and depth of lso	simple	two pairs with the same lso and equal lengths of shortest path will have the same similarity
IC based	The more common information two concepts share, the more similar the concepts are.	Resnik	IC of lso	simple	two pairs with the same lso will have the same similarity
		Lin	IC of lso and the compared concepts	take the IC of compared concepts into considerate	two pairs with the same summation of $IC(c_1)$ and $IC(c_2)$ will have the same similarity
		Jiang	IC of lso and the compared concepts	take the IC of compared concepts into considerate	two pairs with the same summation of $IC(c_1)$ and $IC(c_2)$ will have the same similarity
Feature based	Concepts with more common features and less non-common features are more similar	Tversky	compare concepts' feature, such as their definitions or glosses	take concept's feature into considerate	Computational complexity. It can't works well when there is not a complete features set.
Hybrid method	combine multiple information sources	Zhou	combines IC and shortest path	well distinguished different concepts pairs	parameter to be settled, turning is required. If the parameter can't be turned well it may bring deviation.

Table 2-1: Comparison of Different Semantic Similarity Measures (Meng, Huang and Gu 2013)

2.2 Review and Comparison of Previous Works

There have been many studies and research done on the field of text similarity, and various methods have been used to try and find the similarity between two sentences. The following are some articles reviewed for this project that include various methods for finding similarity between sentences and their findings.

Song et al. (2007) in their paper titled “Question Similarity Calculation for FAQ Answering” proposes a method that takes two sentences, which is the question asked by the user and the question stored in the FAQ database, and calculates the overall similarity by finding the statistic and semantic similarity values of the sentences. In their paper, they mention that the question similarity calculation is the most important stage as it affects the answer quality.

Song et al.’s (2007) method includes the usage of the cosine similarity as the statistic similarity measure and using WordNet to calculate the semantic similarity between two words using their path length between them, followed by calculation of the semantic similarity between two questions using the bipartite mapping by mapping the first question to the second and vice versa. The overall similarity is calculated using the following formula:

$$Sim_{overall} = (1 - \delta)Sim_{statistic} + \delta Sim_{semantic}, \quad (1)$$

Figure 2-2: Overall Similarity between 2 questions

where δ is a constant value between 0 and 1.

According to their experiment, the results obtained shows that a good performance is achieved by using the overall similarity measure as compared to only using statistic or semantic measures. However, as shown in their results using S@n where n=1 performance metrics, the statistical similarity measure gives the lowest result, with 50.0%, followed by semantic similarity measure with 57.1% and the combined similarity measure with 64.3%.

From this, even though combined similarity measure is slightly better as compared to the semantic similarity measure, it is still not good enough when used in a real life application as high recall is required when categorizing hoaxes. Furthermore, it can be seen that semantic similarity is better than statistic similarity, and this contributes to the selection of similarity measurement used in this project.

Achananuparp, Hu and Shen (2008) evaluates various sentence similarity measures whereby the performances of word overlap, TF-IDF and linguistic measures are evaluated and each sentence pair are analyzed with the presumption that they have the same meaning. Their study aims to evaluate the effectiveness of the measures rather than concentrating on estimating the similarity between sentences.

However, according to their article, their semantic similarity measure transforms sentences into feature vectors whereby the feature set is the individual words from a sentence pair. Furthermore, the maximum semantic similarity score between the words in both sentences are only used as term weights and cosine similarity is further added to calculate the sentence similarity. As their method uses the inverse document frequency value in their calculation, a large dataset or corpus is required to calculate the IDF values prior to the actual sentence similarity algorithm.

Furthermore, their study includes word order similarity as another type of sentence similarity measure which focuses on the word order between the two sentences. Combined similarity measures have also been evaluated by combining similarity sentence pair with word order similarity and semantic similarity measure with word order similarity. From the results that they have obtained, linguistic measures that include sentence semantic similarity and combined similarity measures perform significantly better than the others at $p < 0.05$.

They have also proposed using a graph-based representation instead of using a bag of words to represent a sentence. This further supports the usage of Bipartite Graph as in Song et.al's (2007) study. Therefore, Bipartite Graph will be used in this project to get the highest word similarity score between word pairs.

A grammar-based semantic similarity algorithm was proposed by Lee, Chang and Hsieh (2014) for natural language sentences whereby the corpus-based ontology and grammatical rules is proposed. WordNet and grammatical rules is used in the process of representing relationships between pairs of sentences in grammar matrices. For their research, they have used Wu & Palmer's similarity measure (Lee, Chang and Hsieh 2014), and have linked words into subtypes based on their grammar information (nouns, adverbs, adjectives, etc).

According to their paper, their algorithm is the first measure of semantic similarity measure that integrates word-to-word evaluation to grammatical rules, quantifies correlations

between phrases rather than considering word order or common words and that it performs well on sentences similarity and paraphrase recognition (Lee, Chang and Hsieh 2014). Their algorithm is assumed to take a long processing time because linkages between words in a sentence analyzed and further categorized into subtypes for some links.

To summarize text automatically, Aliguliyev (2009) has presented a new sentence similarity measure and sentence based extractive technique in his study. Following this, his paper states that similarity measure plays a role in summarization results besides an optimized function. His method involves the use of sentence clustering, by grouping based on their content or main focus. Furthermore, normalized google distance (NGD) is used, by computing the semantic similarity between concepts thru the number of hits returned by Google whereby labels, which are the concepts, are the input search terms into the search engine.

The experimentation done by Aliguliyev (2009) shows that using the NGD-based dissimilarity measure gives a better performance as compared to the Euclidean distance. However, their use of clustering may not be very accurate as the World Wide Web is very vast, and the sentences that are used as the cluster representative may not correctly represent a certain cluster, which is another factor taken into account in this project, which is to limit the scope only to health-hoaxes to get a more accurate result.

A study was done by Vuković, Pripuzić and Belani (2009) to develop an intelligent automatic hoax detection system using Kohonen's self-organizing maps (SOM) architecture, which is a type of artificial neural network. In their paper, they have emphasized on the importance of pre-processing for text classification and conclude that the proposed system is able to identify and classify hoaxes based on similar patterns.

Their system is automatic, whereby an additional note is added into the title so that users can identify if it is a hoax or not. They have included Croatian besides English for the languages supported, but due to this, they have chosen to use n-gram tokenization instead of stemming or lemmatizing. From their study, it can be further seen that the four most common hoaxes were chained letters that were on prayers, asking for help for a surgery and warning recipients about something, and of these four, three were in Croatian. Therefore, their proposed solution may be more suited towards Croatian than English. Besides that, their method does not take into account the semantic meaning of the email's content, which they hope to develop in the future.

Li et al. (2006) has presented an algorithm that calculates the sentence similarity based on semantic nets and corpus statistics. The overall sentence similarity is calculated by using the semantic information and word order of the sentence with the use of a lexical database (WordNet) and from corpus statistics to give the algorithm adaptability. To further evaluate their similarity measure, they have invited participants to rate the similarity of meaning of sentence pairs used in the research.

The results obtained shows that certain sentence pairs are semantically similar and have also achieved good similarity scores. However, there are some sentence pairs that are not similar, yet achieving high similarity score, and this shows that pre-processing steps such as removing stop words are important and will affect the final similarity score. Furthermore, Li et al. (2006) have stated that the word order vector will only be useful if the “pair of linked words (the most similar from the two sentences) must intuitively be quite similar as the relative ordering of less similar pairs of words provides very little information”.

CHAPTER 3: SYSTEM DESIGN

3.1 Entity Relationship Diagram

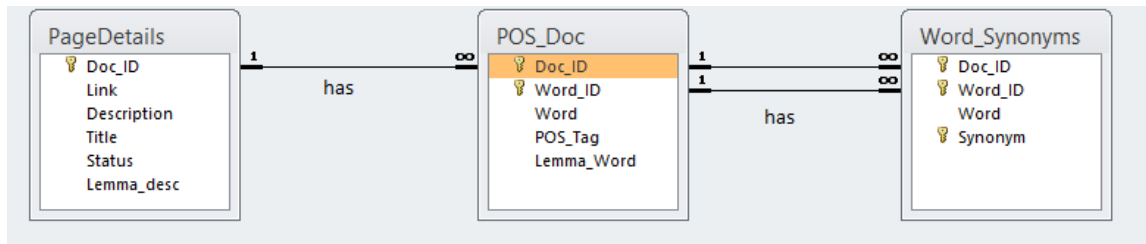


Figure 3-1: Entity Relationship Diagram in the Database

In the diagram above, there are three tables in the database that stores the information needed to calculate the semantic similarity between a highlighted sentence and the description from each collected webpage. PageDetails stores each crawled webpage as a document and assigns the primary key, or the ID, as Doc_ID, along with its link to the webpage, description and title that is obtained from the webpage's header and the status of the webpage (e.g. is it a false or true hoax). The lemmatized description is stored for reference.

In the POS_Doc table, each word in the description in PageDetails is extracted and is given an ID, Word_ID and is stored along with its POS tag, of which would only consist of adjective, nouns and verb POS tags as in the list of POS Tags in Appendix A. The lemmatized word is also stored and is used to compare with the word from the highlighted sentence from the extension when queried from the servlet.

The Word_Synonyms table stores the synonym of each word in POS_Doc. These synonyms are taken from WordNet using the JAWS API and is queried when the servlet queries for a word in the highlighted sentence from the extension.

3.2 Use Case Diagram

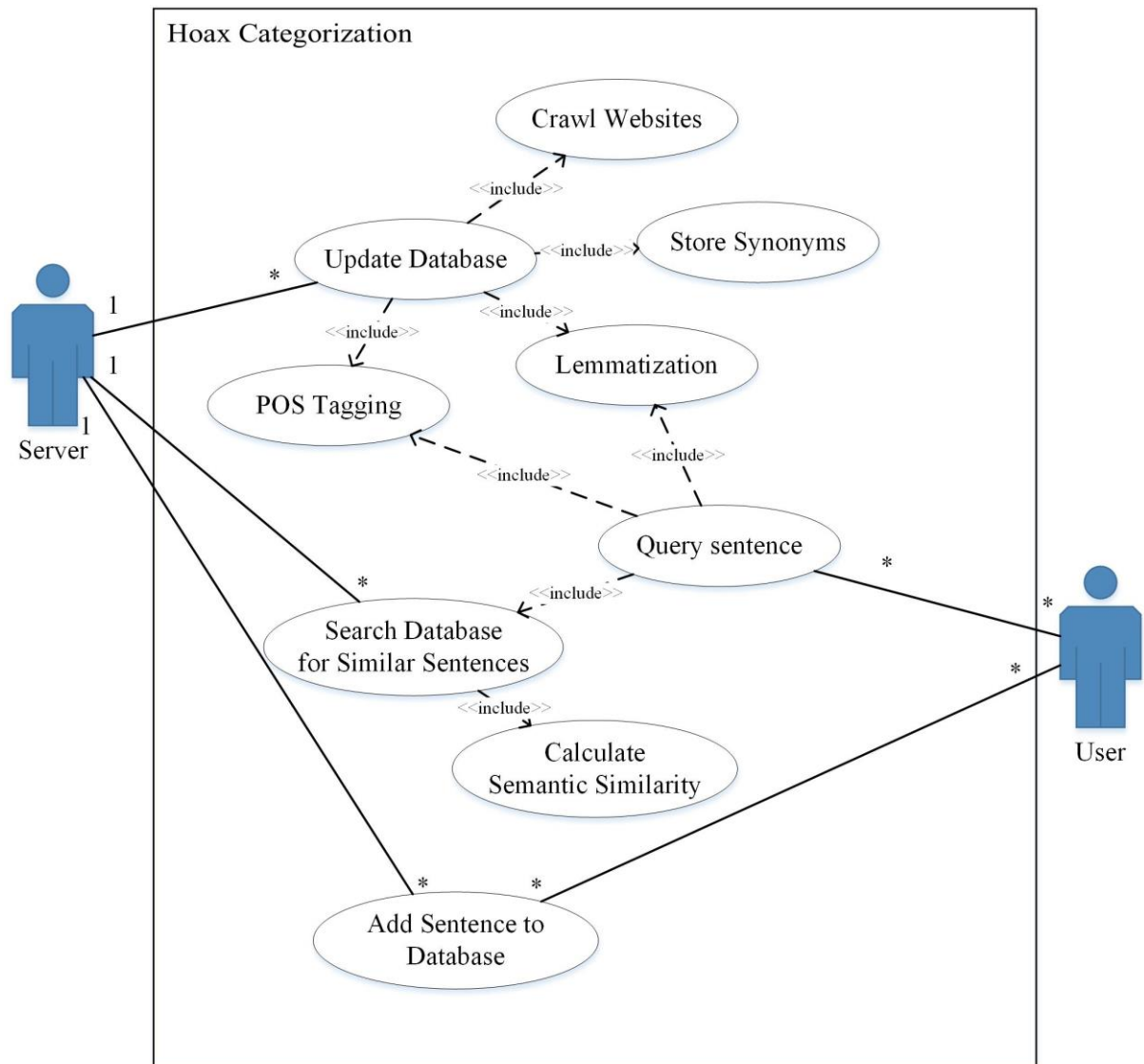


Figure 3-2: Use Case Diagram

In the above use case diagram, it can be seen that the web user queries a sentence, and all other processes are then done within the system. The system does the POS tagging, lemmatization, and searching the database for similar sentences as well as calculating the semantic similarity between the highlighted sentence and these retrieved sentences. The server needs to update the database from time to time, and thus need to crawl the websites again and perform the POS tagging, lemmatization and save the related items into the database.

3.3 Activity Diagram

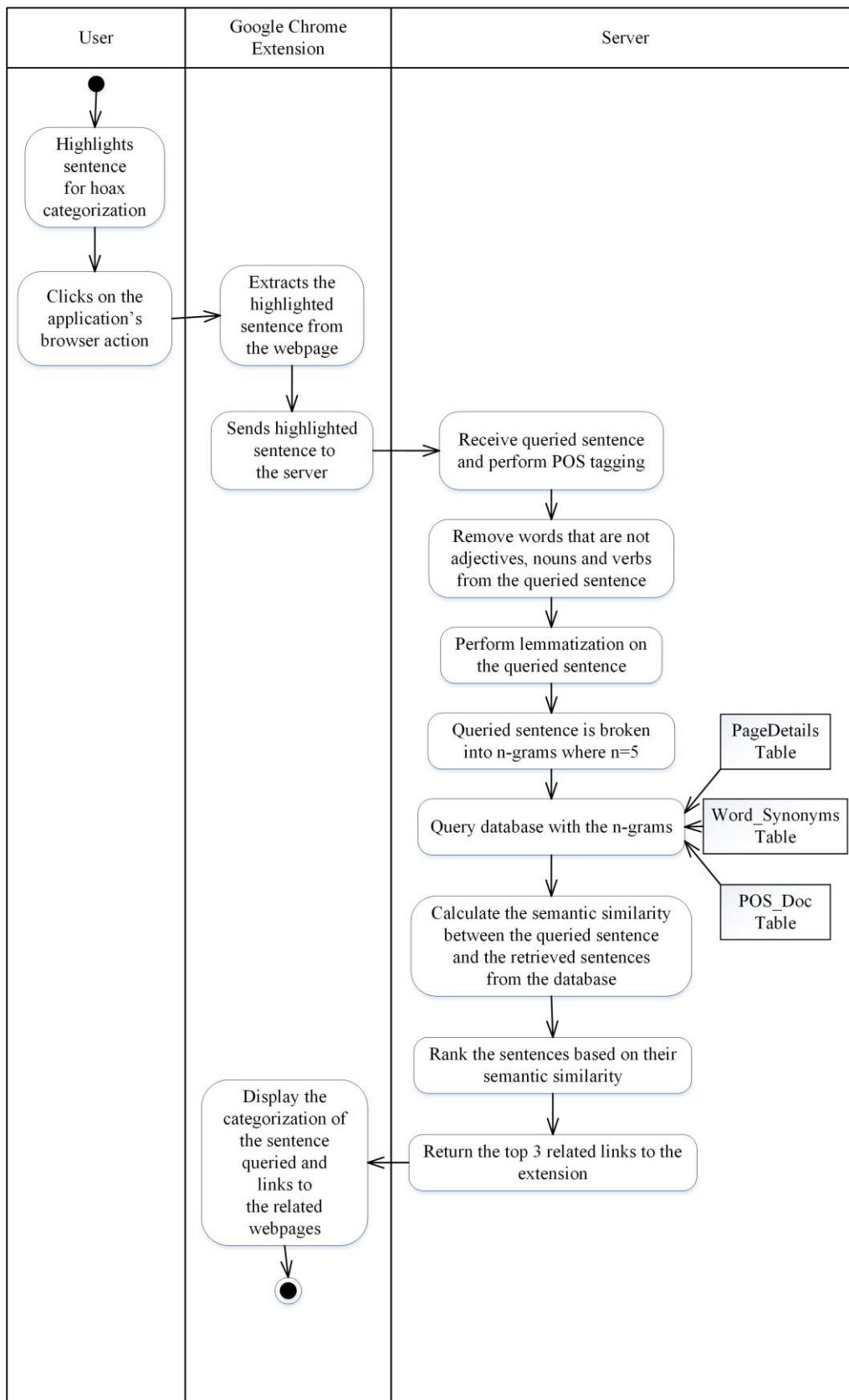


Figure 3-3: Activity Diagram for Google Chrome Extension

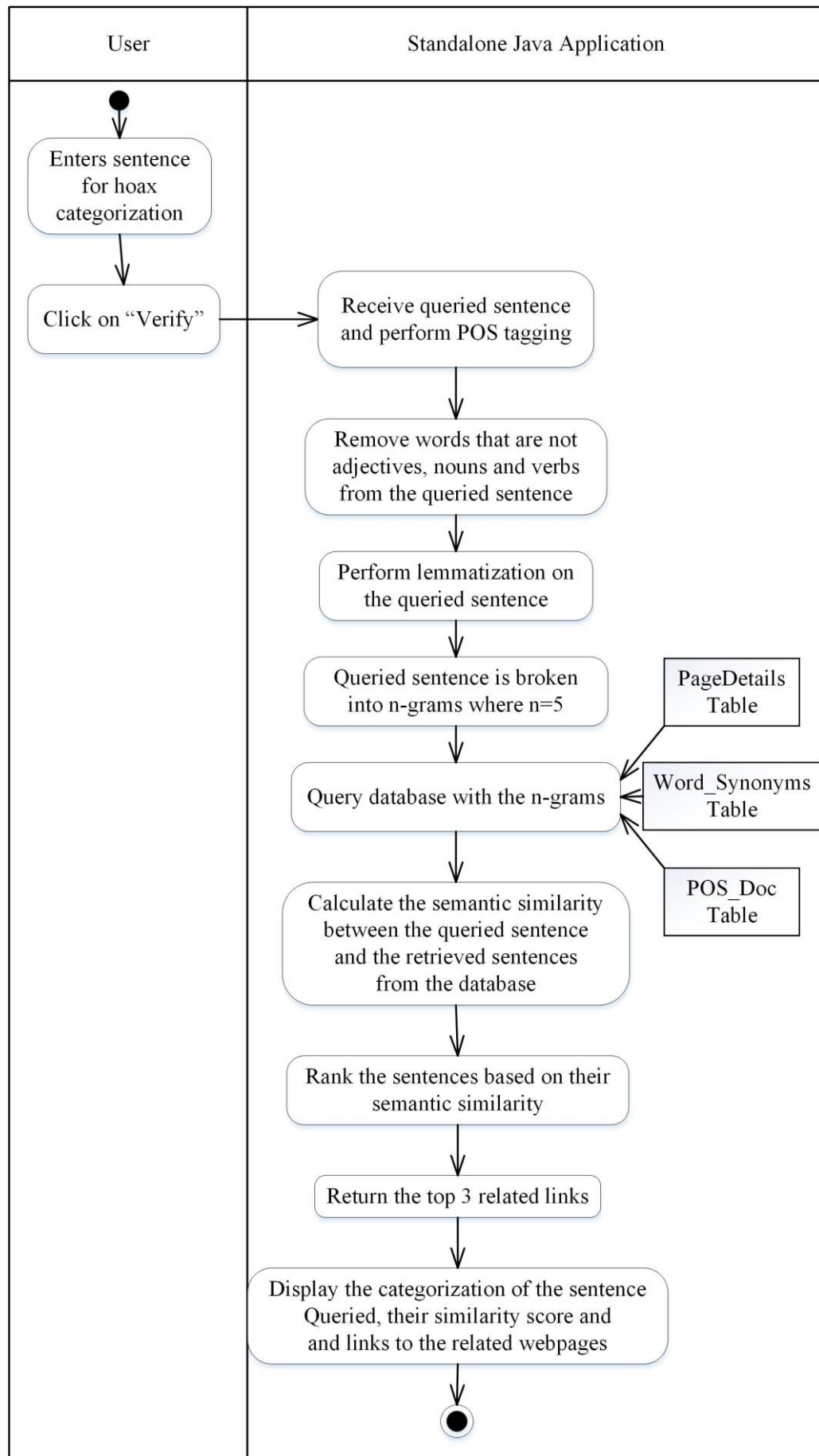


Figure 3-4: Activity Diagram for Hoax Categorization in the Standalone Java Application

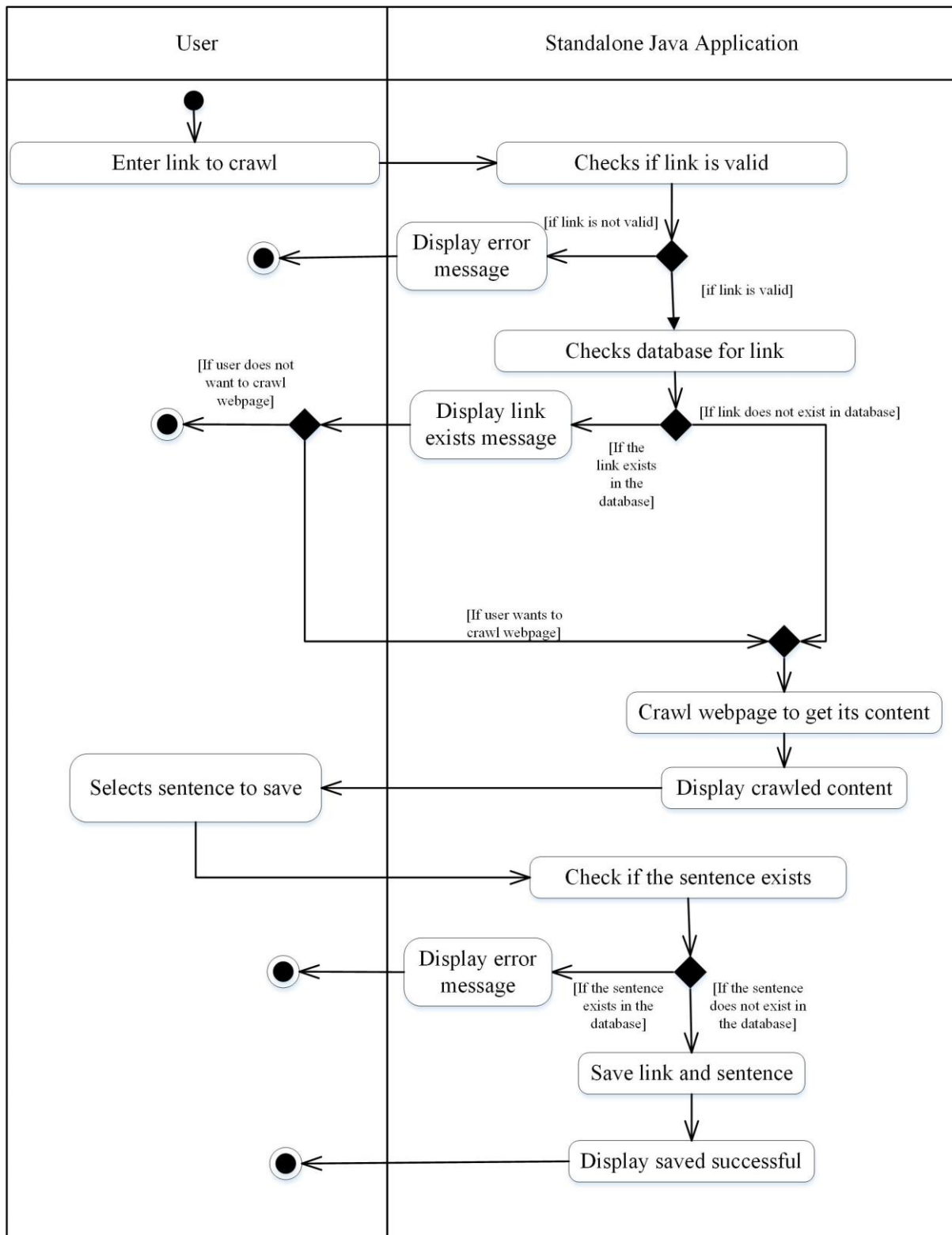


Figure 3-5: Activity Diagram for Crawling Webpage and Selecting Sentences

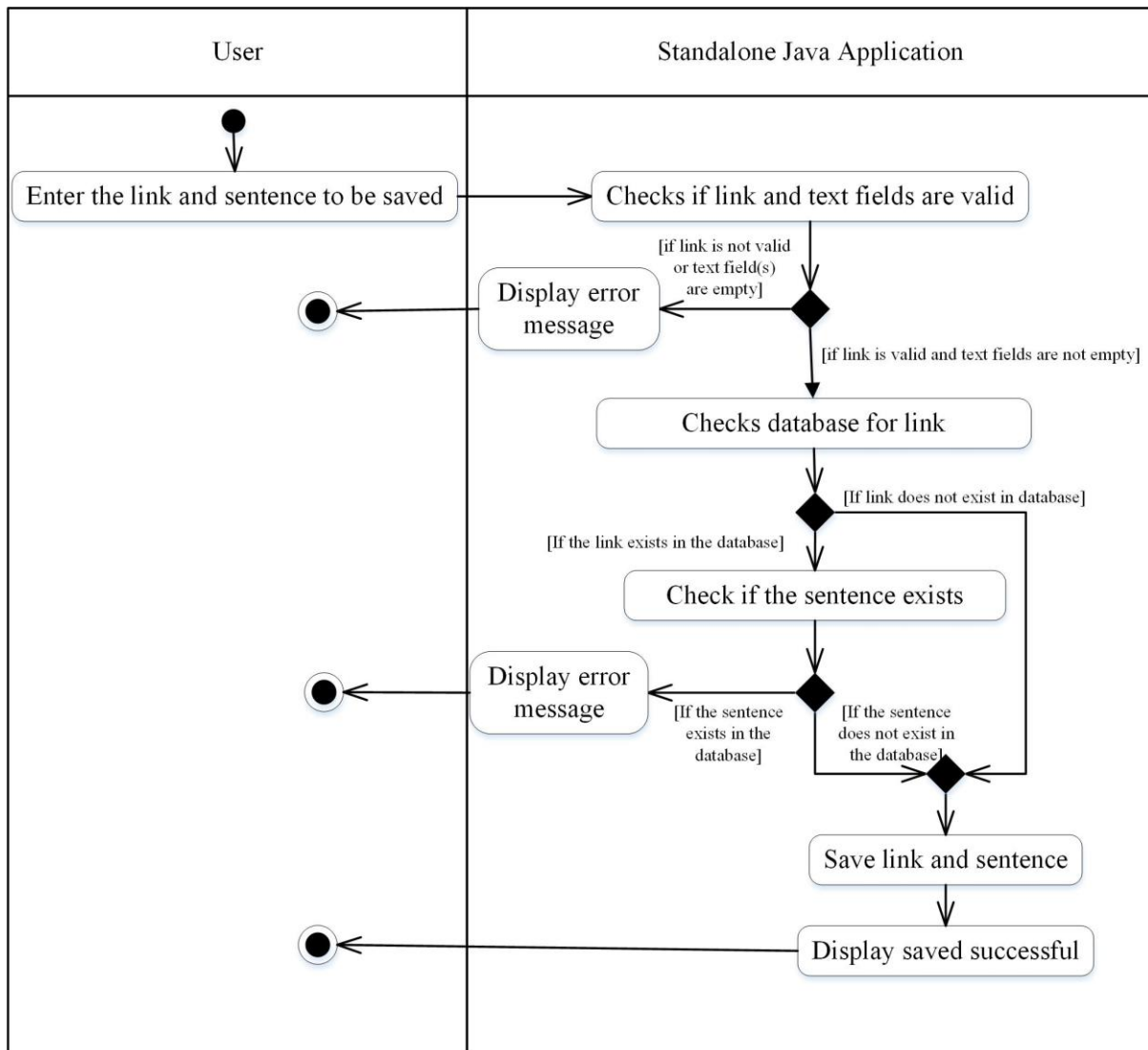


Figure 3-6: Activity Diagram for Saving Link and Sentence Only

CHAPTER 4: METHODOLOGY, TOOLS AND SYSTEM REQUIREMENTS

4.1 Methodology

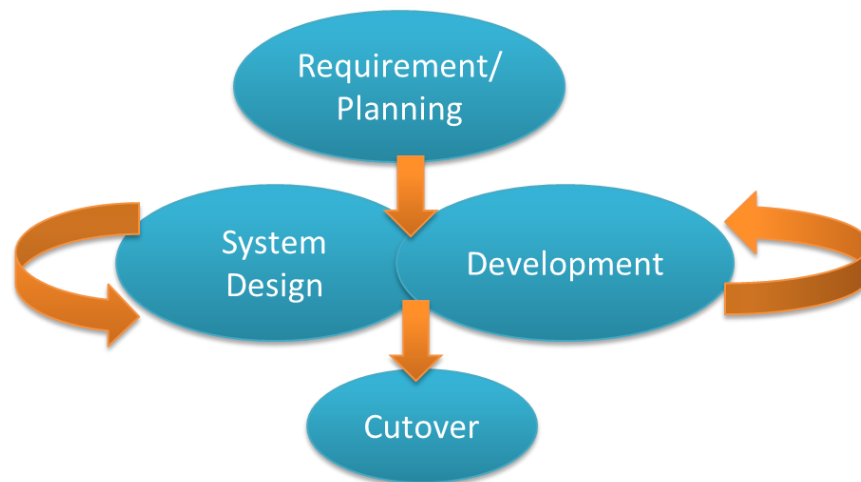


Figure 4-1: Rapid Application Development Methodology (Javatechig | Resources for Developers 2012)

For this project, the best methodology to adopt would be the Rapid Application Development (RAD) approach. RAD is a methodology for development of systems to reduce design and implementation time drastically, and heavily rely on users' involvement and prototyping. This methodology is most suitable to develop the extension as it uses web technologies and is dependent on various other software such as Google Chrome and Eclipse and the due to the rapidly changing technology that may cause the system to be obsolete if it were to be developed using traditional methodologies such as the traditional waterfall Systems Development Life Cycle (SDLC). In addition, RAD allows feedback from the users during the development stage which is very important to ensure that the outcome that is analyzed by the extension is accurate and fulfills the purpose of the application.

RAD life cycle contains 4 phases: Requirements Planning, User Design, Construction and Cutover. These phases are shorter and are combined so that a streamlined development technique can be obtained. This project would concentrate on the usability (its system function) and the user interface requirements based on the system performance and function needed from the system itself. Since the application has to be produced in a short amount of time, RAD removes the time-consuming activities of comparing with existing standards and systems during development and design,

RAD focuses on the design and development phases, which helps to ensure the application produced is what the user wants and need. And since time between the end of design and implementation is shorter, it can be certain that the system is closer to the current needs of the user and therefore the application produced is of higher quality if compared to those developed in the traditional way.

Prototype is created during the user design phase, which allows early testing prior to the development of the application itself. The testing and system documents are created during the construction/ development phase, which is because of the iterative development process that heavily depends on the feedback of users. The most important deliverable, the application itself, is produced during the development phase and is constantly modified between the iterative process of design and construction phases.

4.2 Tools Used

For this project, the following tools and software will be used:

4.2.1 Programming Languages

4.2.1.1 Java

Java is the selected language used for crawling, preprocessing and ranking processes. This is because Java is a widely used programming language and various tutorials are provided to assist in building programs needed for these processes. In addition, Java imports external libraries easily, which helps in the development of the programs for these processes. Some of the imported libraries are as follows:

a. Crawl4j ver. 3.5

Crawl4j is an open source Java crawler that can be obtained via <https://code.google.com/p/crawler4j/>. The purpose of using this crawler is to crawl through websites such as www.hoax-slayer.com and www.webmd.com to collect hoax and non-hoax data to be stored in the database.

b. Stanford CoreNLP ver 3.5.1

Stanford CoreNLP is an open source software by Stanford NLP (Natural Language Processing) Group at Stanford University that is available on their website <http://nlp.stanford.edu/software/index.shtml>. It is used to lemmatize the words in each document that was crawled to try and reduce the word lists by changing words into a form of which its meaning is still intact, for example: “buy” and “buying” refer to the same meaning, only in different forms. Lemmatizing takes into consideration of the sentence to decide the context of the word and what is its simplest form without derailing from its original intended meaning.

c. Stanford Log-linear Part-Of-Speech Tagger

Stanford Log-linear Part-Of-Speech Tagger is another open source software by Stanford NLP (Natural Language Processing) Group at Stanford University that is available via <http://nlp.stanford.edu/software/tagger.shtml>. The purpose of this library is to only categorize each word to a part of speech, such as nouns, verbs and adjectives. Despite having a full version that includes models for Arabic, Chinese, French, German and Spanish, only the package with the English trained model is used for this application.

d. Java API for WordNet Searching (JAWS)

JAWS is an API that is used to find synonyms of a particular given its part-of-speech. These synonyms are then stored in the database so that the search boundary is widened when searching for similar sentences from the database. The API is available from <http://lyle.smu.edu/~tspell/jaws/>.

e. UCanAccess ver 2.0.9.3

Java 8 no longer supports the JDBC-ODBC Bridge, and therefore an external API is needed to access to the Microsoft Access database. It is easy to implement as the .jar file is imported into Eclipse and the existing codes used can be retained and is available via <http://ucanaccess.sourceforge.net/site.html>.

f. Java Development Kit (JDK)

JDK is required prior to the installation of Eclipse and Apache Tomcat as the development is done in Java.

4.2.1.2 Javascript

Javascript is a widely used web programming language and is used to write scripts needed in the extension, such as collecting the highlighted sentence from the webpage, communicating with the server via sending and receiving data and displaying the received results to the user.

4.2.1.3 Hypertext Markup Language (HTML)

HTML is used to design the extension popup page to display to the user the current status of the extension, for example, informing the user that the extension is still awaiting results from the server and displaying the top 3 links with the highest similarity score with the categorization result to the user.

4.2.2 Crawling, Preprocessing and Ranking of Similar Links

Eclipse is a Java Integrated Development Environment that is available for download on their website <https://www.eclipse.org/downloads/>, and is a widely used Java IDE. Previous knowledge and experience on using this IDE contributed to its usage as well.

4.2.3 Development of extension – Google Chrome Extension

Google Chrome is the chosen browser to implement an extension to demonstrate and display the application of the categorization in the real world. Development of extensions on Google Chrome is based on web programming languages such as

Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) for styling and Javascript, client-side programming as well as JQuery, a library for Javascript.

4.2.4 Server

The chosen server to for the prototype to connect to is Apache Tomcat. This is because it is open sourced, and is easily integrated into Eclipse IDE. It is available via their website <http://tomcat.apache.org/>. In addition, Java programming language is used to develop servlets to send and receive data to and from the extension.

4.2.5 Lexical Database

Wordnet is an English lexical database that is used to calculate the semantic similarity between two words. The latest version for Windows is Wordnet 2.1 and is available from <https://wordnet.princeton.edu/wordnet/download/>. After installation, it is an executable program that has a graphical user interface to search for words and their senses and gloss. However, to access this database, the usage of various API's are needed in order to fully utilize and retrieve data from it.

4.2.6 Database

The database used in this project is Microsoft Access 2010 as it is already available to use during development and has a graphical user interface that is easy to navigate around. Prior knowledge on the software also contributed to its selection as the database for this project.

4.3 System Requirements

The end user would require a stable Internet connection and is encouraged to install the latest version of the Google Chrome browser, version 41.0.2272.89 m (as of 19th March 2015) and according to the Google website (Support.google.com, n.d.), the following is system requirements for Google Chrome's optimal performance:

	Windows requirements	Mac requirements	Linux requirements
Operating system	<ul style="list-style-type: none"> • Windows XP* Service Pack 2+ *supported until April 2015 ↗ • Windows Vista • Windows 7 • Windows 8 	Mac OS X 10.6 or later	Ubuntu 12.04+ Debian 7+ OpenSuSE 12.2+ Fedora Linux 17
Processor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Free disk space	350 MB		
RAM	512 MB		

Figure 4-2: System Requirements for Google Chrome Browser (Support.google.com, n.d.)

Since this project utilizes the client-server architecture, the server-side would need to install Apache Tomcat and the Eclipse IDE. Since Eclipse and Tomcat both require Java Development Kit installed prior to installation, therefore according to Oracle (Docs.oracle.com, n.d.), Java runtime requires a minimum of 128MB of memory while disk space requirements for the development tools for Windows 64-bit operating systems is 181MB, source code takes up 27 MB while the JavaFX SDK take 68MB and the runtime 32MB. The recommended operating system would be the operating systems used during the development and testing stages, which is Windows 8.1 and lastly, a stable internet connection is required for the sending and receiving of data to and from the client.

CHAPTER 5: SPECIFICATIONS, IMPLEMENTATION AND TESTING

5.1 System Performance Definition

Precision and recall are measures which are widely used in Information Retrieval (Aliguliyev 2009). Therefore, to evaluate how the extension is performing, the number of test records correctly and incorrectly classified by the system is counted and are tabulated as in the confusion matrix as shown below:

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

Figure 5-1: Confusion Matrix for Tabulation of Two-Class Classification Results and the Various Performance Metrics that can be Calculated (Chuah 2014)

To evaluate the performance of the system, the precision and recall of the system will be analyzed. To obtain the overall recall of the system, the log scale fitness is used to measure the recall of each sentence given the number of sentences retrieved and the position of the actual sentence amongst the sentences retrieved by scaling them to the log function. The equation is as follows:

$$recall_n = \log(N + 0.5) - \log(n)$$

Where N is the number of retrieved sentences for a particular sentence, and $n = 1, 2, \dots, N$ is the ranking of the retrieved sentences after performing the sentence similarity. $recall_n$ is the recall rate for the n^{th} ranked retrieved sentence.

The above equation will then result in an array of values, which is then scaled by finding the maximum value of the $recall_n$ in the N retrieved sentences and dividing each

$recall_n$ value with that maximum value so that the maximum value of the entire array is 1.0. The precision of the particular sentence is given as the $recall_n$ value where n = the actual sentence ranking position. So for example, if the first result in the retrieved sentences is the actual sentence with which the testing sentence is from, then the recall value of that particular sentence is 1.0. If the actual sentence is not among the retrieved sentences, the recall rate is known to be 0. Lastly, the overall recall rate is as follows:

$$recall_{overall} = \frac{\sum_{i=1}^N recall_i}{N} \times 100\%$$

Where N is the number of sentences tested (size of testing data), and $\sum_{i=1}^N recall_i$ is to sum up all the recall rates of each sentence tested.

To find the precision value of the system, the number of times the actual sentence is ranked as the first among the retrieved sentence divided by the total number of sentences tested (size of testing data) multiplied by 100% to get the percentage.

5.2 User Interface Design

There are two different applications that are produced at the end of the project duration: a Google Chrome extension and a standalone Java application. Both applications aim to provide similar functionalities.

5.2.1 Standalone Java Application

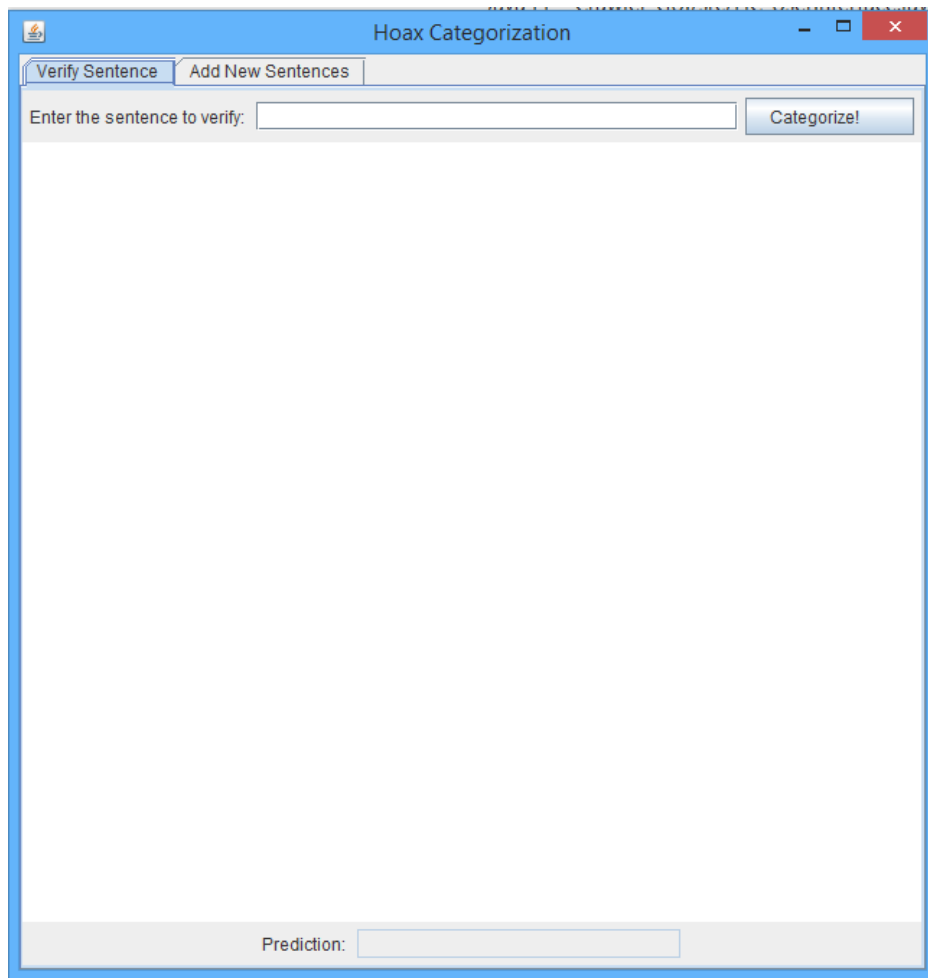


Figure 5-2: Tab for Verifying a Sentence in the Standalone Application

The above figure shows the first screen displayed when the application is run. The user is allowed to choose between the functions of verifying a sentence if it is a hoax or not, or to add new hoax/non-hoax sentences to the database. Clicking on the Red Close Button “X” will close and terminate the program.

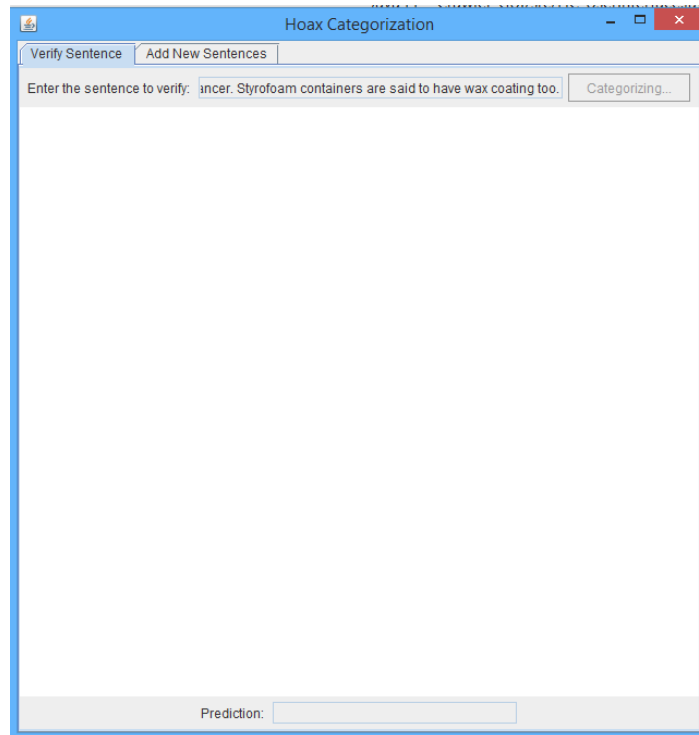


Figure 5-3: Categorizing a sentence in the Standalone Application

In this screen, it can be seen that the button and text field for the sentence has been disabled when the system is currently categorizing a sentence. During this time, the user can explore the “Add New Sentences” feature.

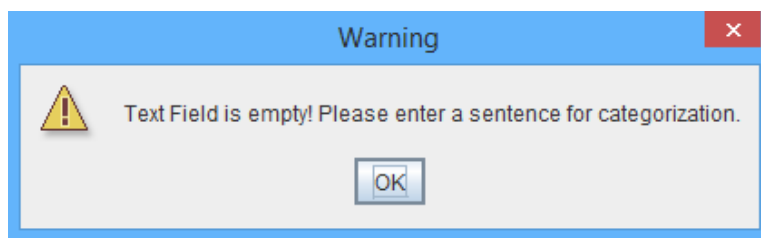


Figure 5-4: Popup to inform user that there was no sentence entered

If the user clicks on the “Categorize” button without entering any sentence to categorize, a warning will pop up as above.

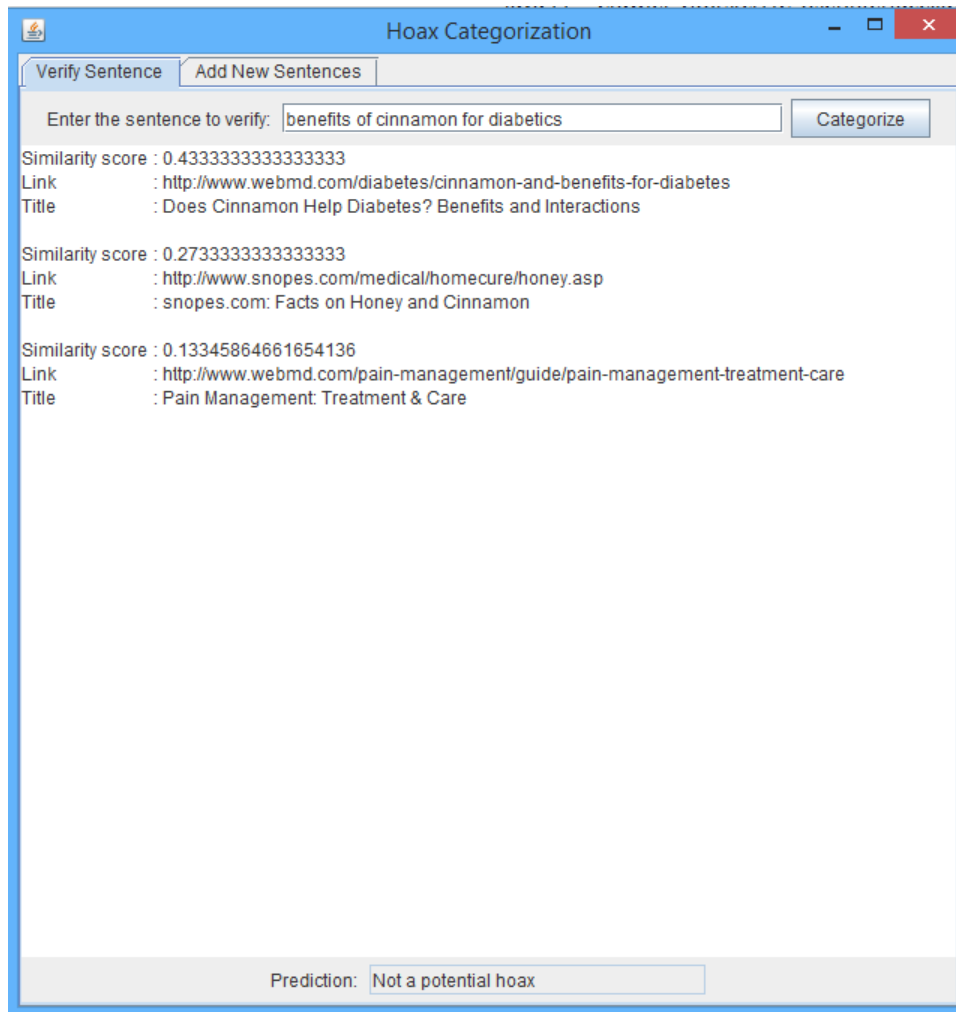


Figure 5-5: Displaying Categorization Results in the Standalone Application

Similar to the Google Chrome extension, the standalone Java application also retrieves and displays the top 3 links that are related to the sentence specified by the user. The prediction whether or not it is a hoax is displayed in the text box at the bottom of the window as shown above.

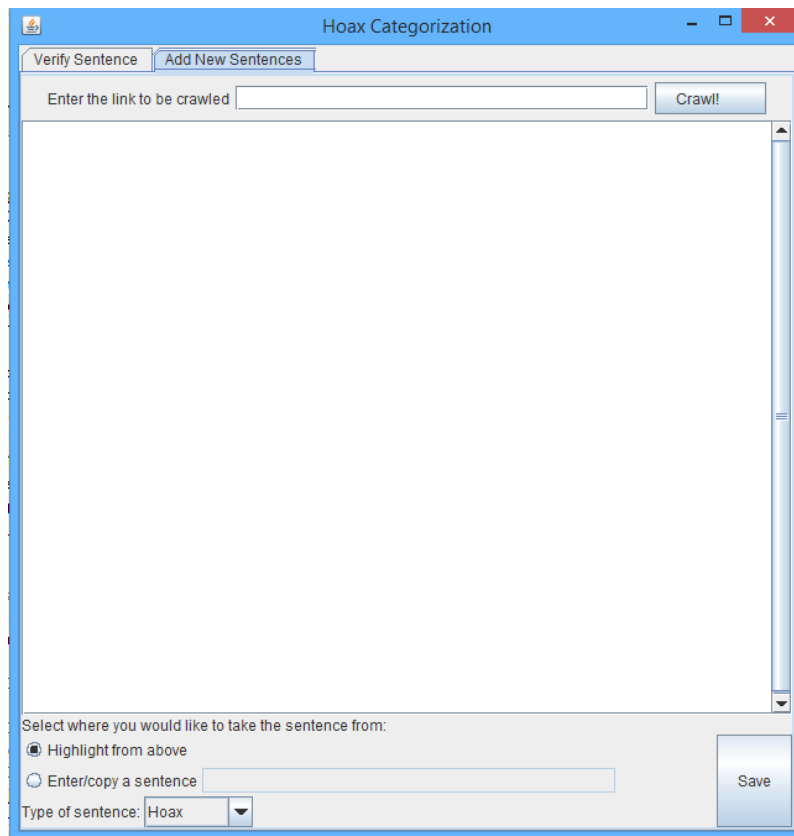


Figure 5-6: Screen For Adding New Sentences

Users can insert new sentences in two ways, the first is the user inputs a webpage link where he/she would like to get the sentences from while the second is where the user directly enters the webpage link at the top, and fill in the sentence text field in the bottom, along with its specified type (e.g. hoax or non-hoax) and finally clicking the “save” button to store in the database.

There are cases where the link provided already exists in the database, and as such the system will show the following pop up to ask if the user would like to proceed with the crawling or not.

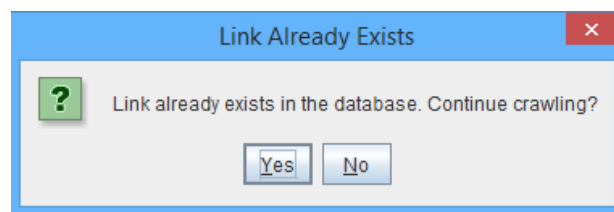


Figure 5-7: Popup informing the user that the link exists in the database

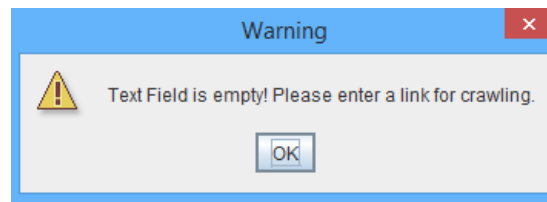


Figure 5-8: Popup informing the user that no URL was entered

If in the event that the user did not provide any URL to the webpage for crawling, the above popup will appear. Similarly, the system also checks to see if the URL entered is valid, else the following popup message will be shown.

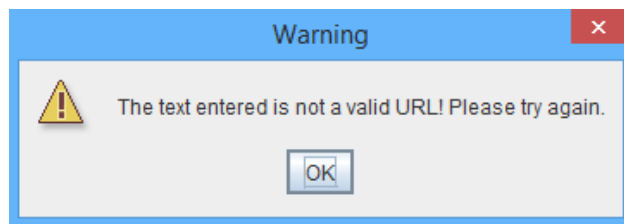


Figure 5-9: Popup informing the user that the URL entered is not valid

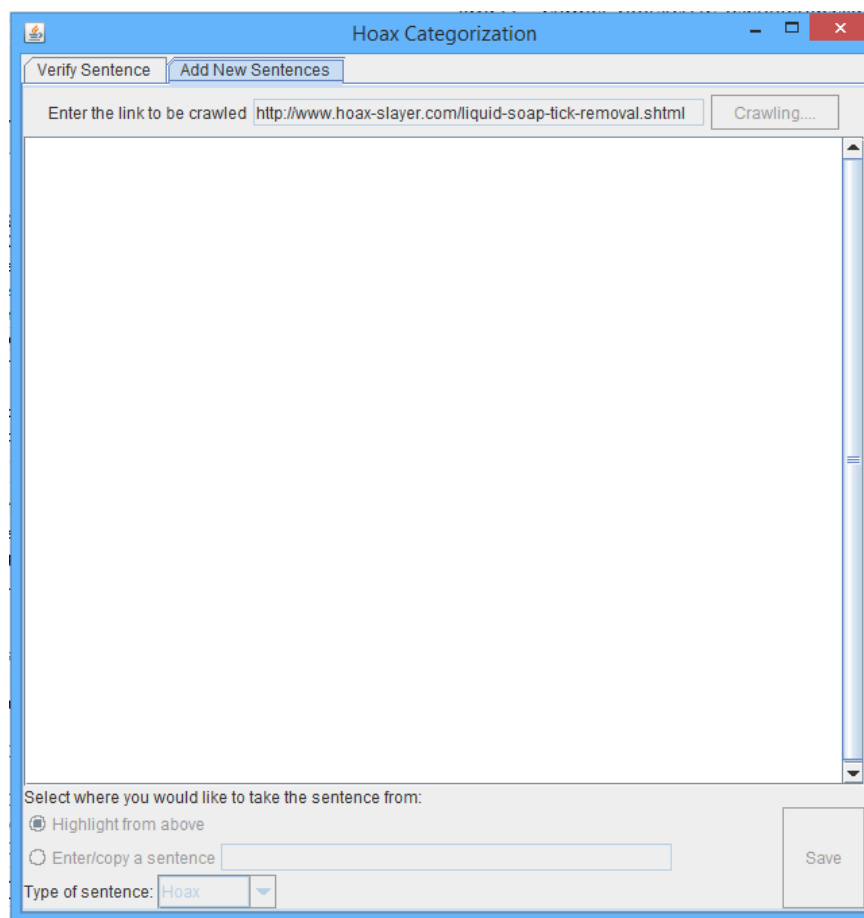


Figure 5-10: Screen when crawling the webpage

After the user enters the link to crawl, the above screen will be shown, whereby the “Crawl” button changes to “Crawling...” and the features for saving are also disabled to prevent the user from entering values until the page is crawled have returned the content of the page. The user is free to verify a sentence while the crawling is still running.

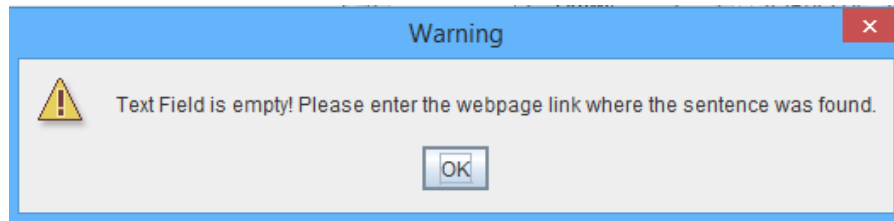


Figure 5-11: Inform user that sentence cannot be saved without the link

The above message dialog pops up when the user attempts to save a sentence without providing the link to the sentence.

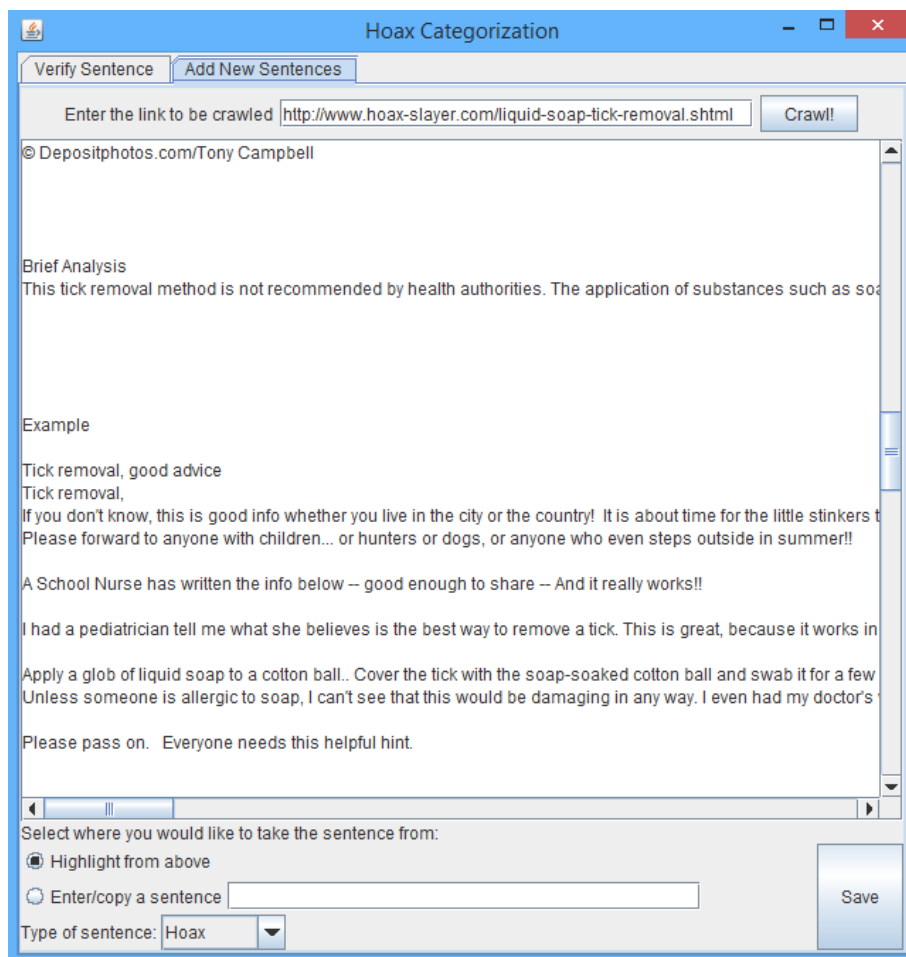


Figure 5-12: Screen after crawling is successful

The previous screen is shown after crawling has been successful. The features for saving or crawling are re-enabled which allows the user to proceed to what he/she would want to do next.

There are two options that the user can choose to save the sentence(s). The saving process can only handle a sentence at a time, and the user would need to manually insert them at his/her own discretion of where the sentence should start and end. The two options include either highlighting from the crawled content, copy and pasting into the text field for the sentence, or typing directly into the text field. The following message is shown once the sentence is successfully saved in the database.

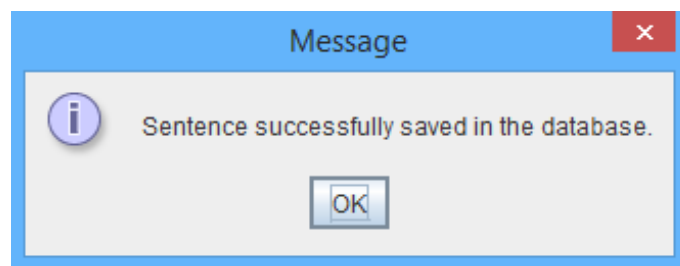


Figure 5-13: Successfully saved sentence popup message

There are cases where the link already exists in the database, but the user would like to add in another sentence that references to that link. If the sentence is successfully saved, the following message will show:

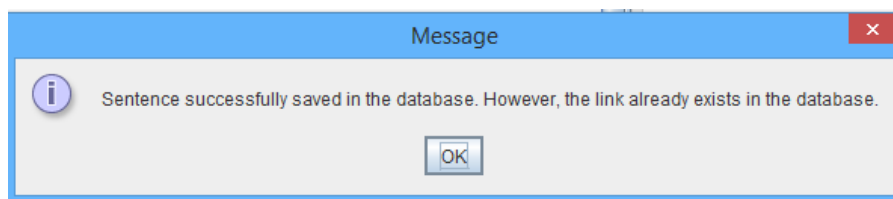


Figure 5-14: Popup to state that the sentence has been successfully saved with existing link

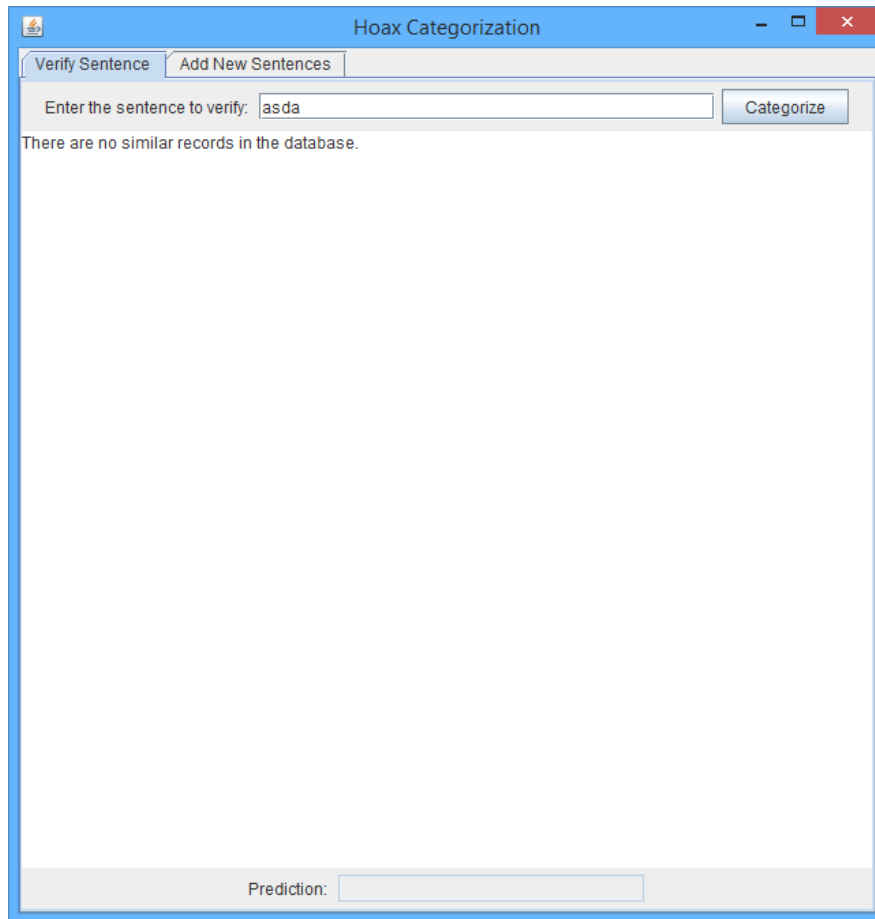


Figure 5-15: Screen if there are no similar records in the database

The above screen will be displayed if there are no similar records found in the database, and therefore, no sentence similarity can be done.

5.2.2 Google Chrome Extension

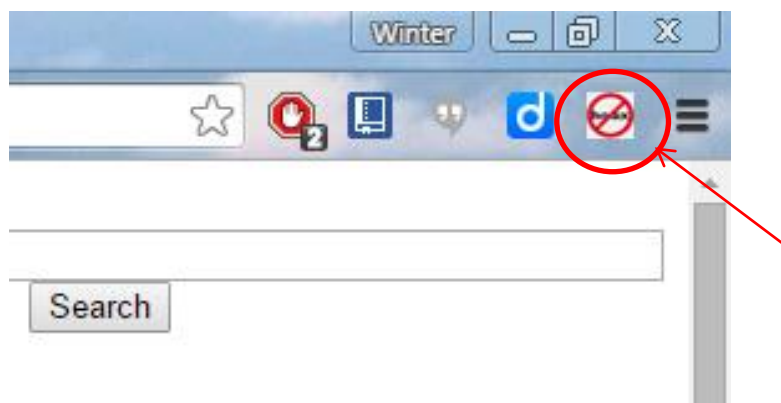


Figure 5-16: The Browser Action icon of the Google Chrome Extension

The figure above shows the browser action icon for the Google Chrome Extension. The user can click it from any webpage.

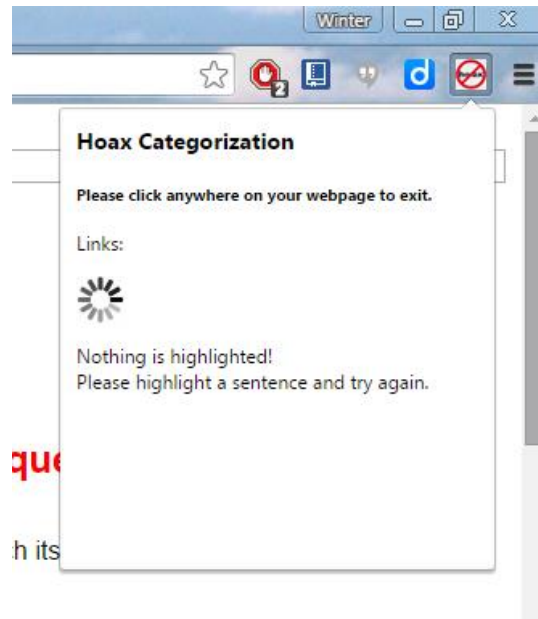


Figure 5-17: Screen if there's no sentence highlighted

The extension works by extracting out the sentence that the user has highlighted in the web page. If he/she did not highlight any sentence, the above screen will be shown. However, if the user has highlighted a sentence, then the following screen will be displayed.

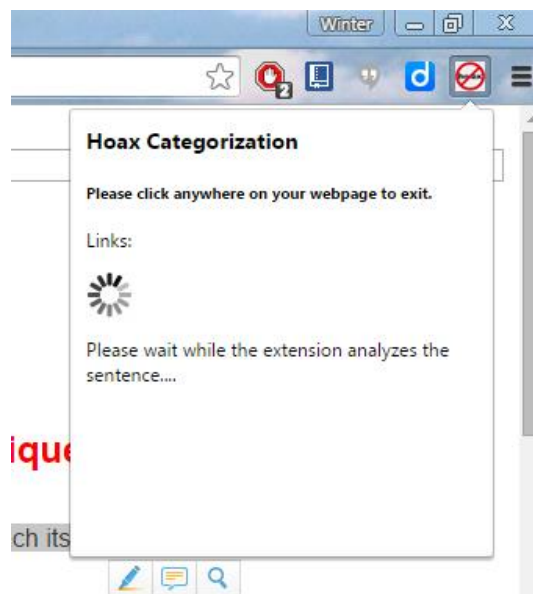


Figure 5-18: The extension has sent sentence to database and awaiting response

Once the server respond with the links, titles and prediction, the following screen will be displayed.

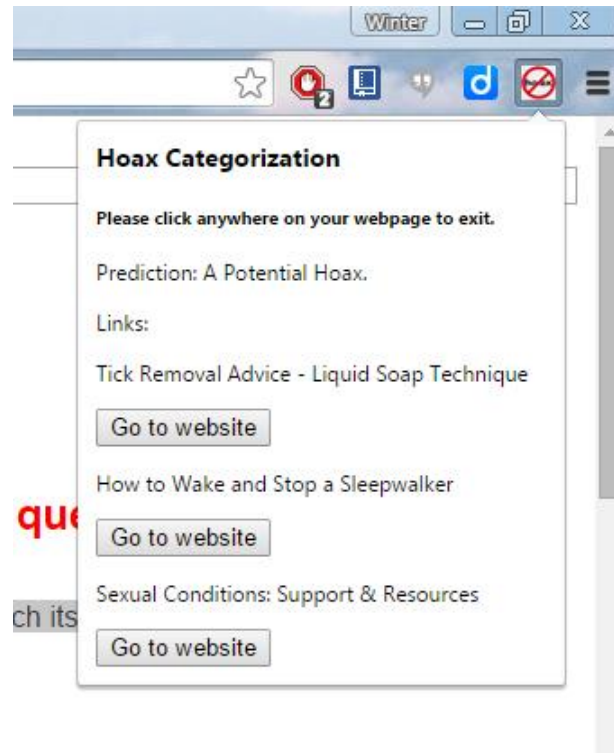


Figure 5-19: The Related Links to the Highlighted Sentence

The above display is shown once the server finishes analyzing and the user can click on the button below the related link title to go to the website. A new tab will be opened that links to the website selected.

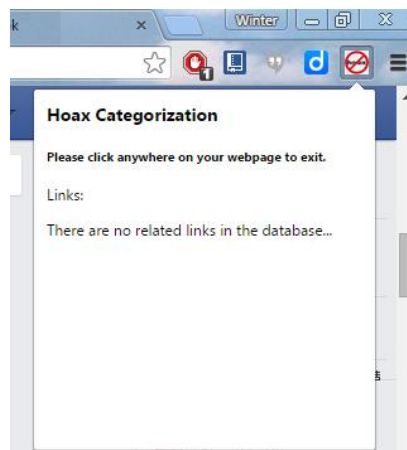


Figure 5-20: Screen when there are no related links in the database

The above screen will be displayed whenever the user highlights a sentence that the database does not have any related links to, or in a different language such as Korean, Japanese and Chinese.

5.3 Verification Plan

To obtain the optimum n value when querying the database using n -grams, the value of n is found using distributional methods. Firstly, the number of words of each sentence in the database was tabulated and a graph is produced. Based on this graph, the median of the data set is chosen as it is a right-skewed graph. The n value is now narrowed down from 1 to the median number. Finally, a few sentences was chosen and the number of retrieved sentences when $n = 1, 2, \dots, 10$ is recorded to see the optimum value of n where the number of records returned are not too many or too few.

To ensure that the developed extension gives the desired outcome to fulfill the objectives of this project, systems-level black box testing is done. Black box testing tests the system without seeing or having any knowledge of the internal structure or codes. The testing is done from the user's point of view and only knows what to expect of the outcome without any knowledge of what goes on behind the scenes. As the extension does not involve many modules, and has a simple user interface, this testing method is the most suitable to observe if the systems works as it should be.

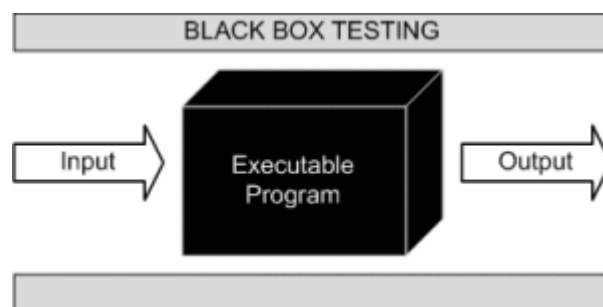


Figure 5-21: Black Box Testing (Softwaretestingfundamentals.com 2010)

There are many advantages of using the Black Box Testing, such as the tester need not be a technical person and is done from the user's point of view, can be used to check the weaknesses of the system, and the test cases can be designed after the completion of functional specifications (Softwaretestinghelp.com, n.d.; Softwaretestingfundamentals.com 2010). Some of the disadvantages include the small amount of test cases generated which still gives the system some weak points and the difficulty in producing test cases as it is hard to identify all the possible inputs within a limited time (Softwaretestinghelp.com, n.d.; Softwaretestingfundamentals.com 2010).

As the input for the system is mainly only text, therefore, the black box testing's error guessing method is used to generate input for the test cases. Error Guessing attempts to find

where the errors can be hidden and therefore there are no specific tools for this method and the test cases written should try to cover as many areas of the application as possible (Softwaretestinghelp.com, n.d).

5.4 Testing Results

To obtain the optimum value of n for the n -gram, the number of words in a sentence and the number of sentences that have the same number of words are calculated. The results obtained are in Appendix B and a summarization of the results is shown in the chart as follows:

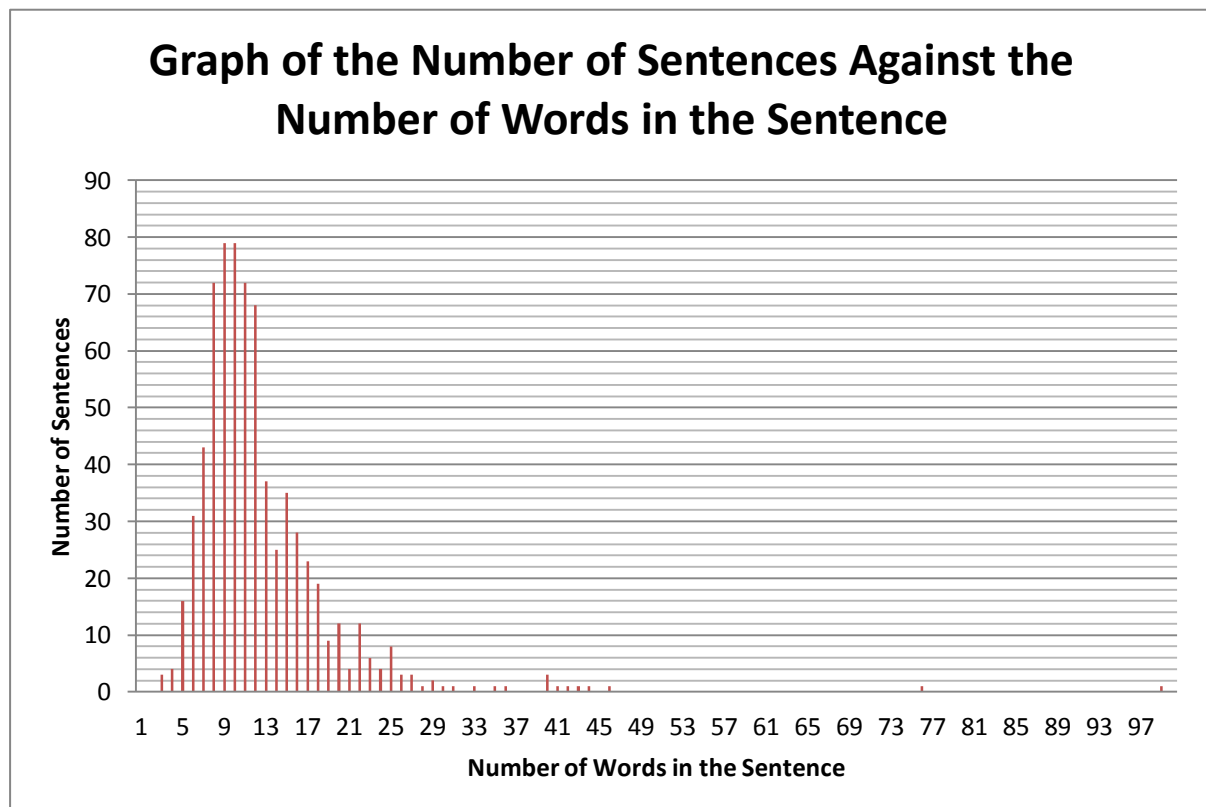


Figure 5-22: Graph of the Number of Sentences against the Number of Words in the Sentence

Based on the graph and results obtained as in Appendix B, majority of the sentences in the database contains ranges between 4-24 words. It can also be seen that there are outliers, such as the one sentence that only 98 words. Therefore, based on the distribution as seen above, the mode and mean would not give a very good estimation of the values of n to be used for the n -grams. The median, which is the middle score, would be better for describing the typical value (Ltcconline.net, n.d.) as it is not affected by outliers.

The median takes the middle number of the entire distribution, where in this distribution that has 713 sentences that has 2 to 98 words, the median would follow the following equation:

$$\text{median} = \frac{713 + 1}{2} = 357^{\text{th}} \text{ sentence}$$

Thus, as highlighted in Appendix B, the number of words where the median lies is 10. Therefore, when finding for the appropriate value of n to be used in the n -grams, the values of n that will be tested is from 1 to 10.

Using the value obtained above, where $n=10$, five sentences with a word length of 10 after lemmatization are used to obtain the number of intersected sentences using n -grams where $n=1, 2, 3, \dots, 10$. The detailed results are shown in Appendix C and the summarized tables are shown below:

Sentence 1:

“Message claims that a study found potentially harmful bacteria on lemons slices used in restaurants.”

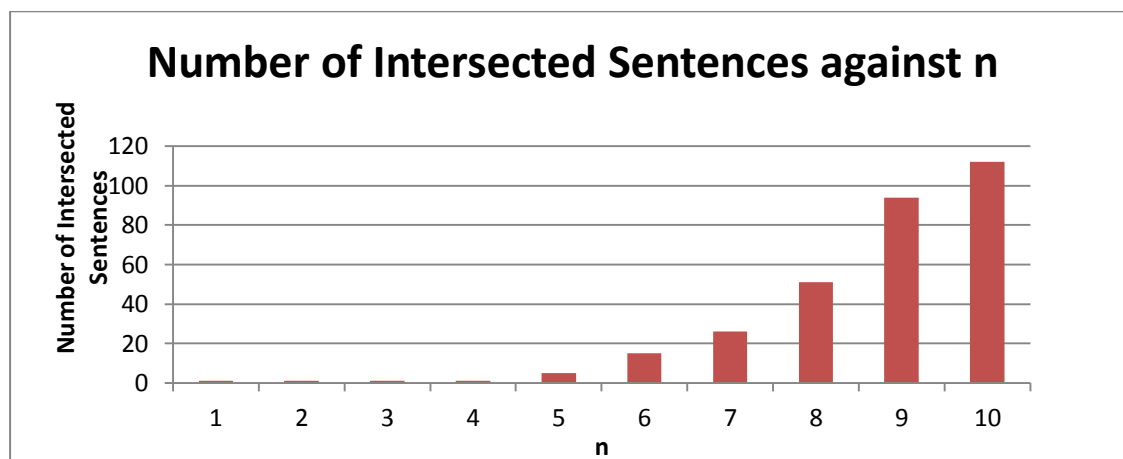


Figure 5-23: Graph of the Number of Intersected Sentences against n for Sentence 1

Sentence 2:

“Researchers found patients were more likely to be informed only after their disease had advanced.”

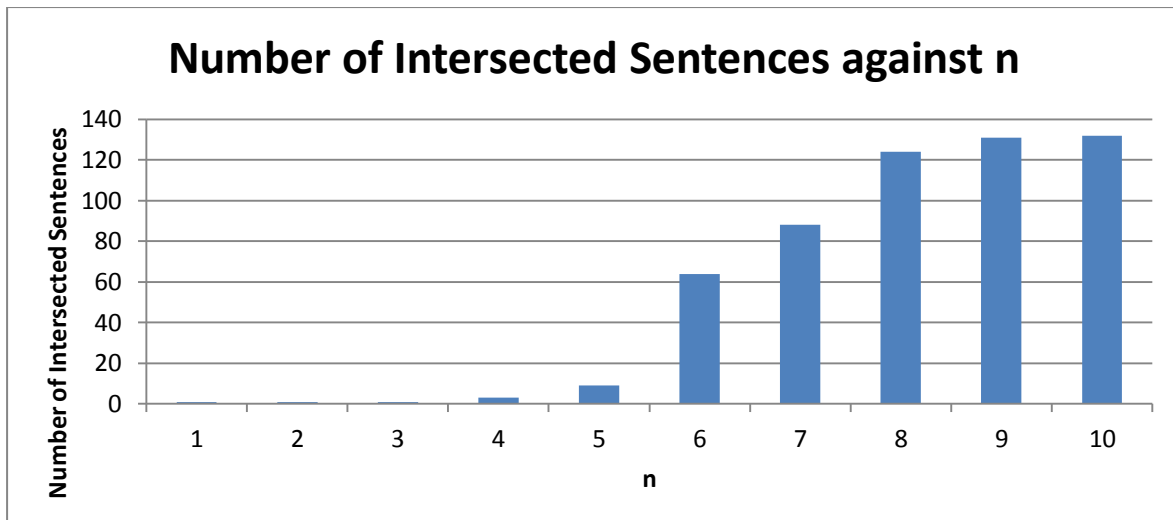


Figure 5-24: Graph of the Number of Intersected Sentences against n for Sentence 2

Sentence 3:

“Ankylosing spondylitis is a form of arthritis that is long - lasting (chronic) and most often affects the spine.”

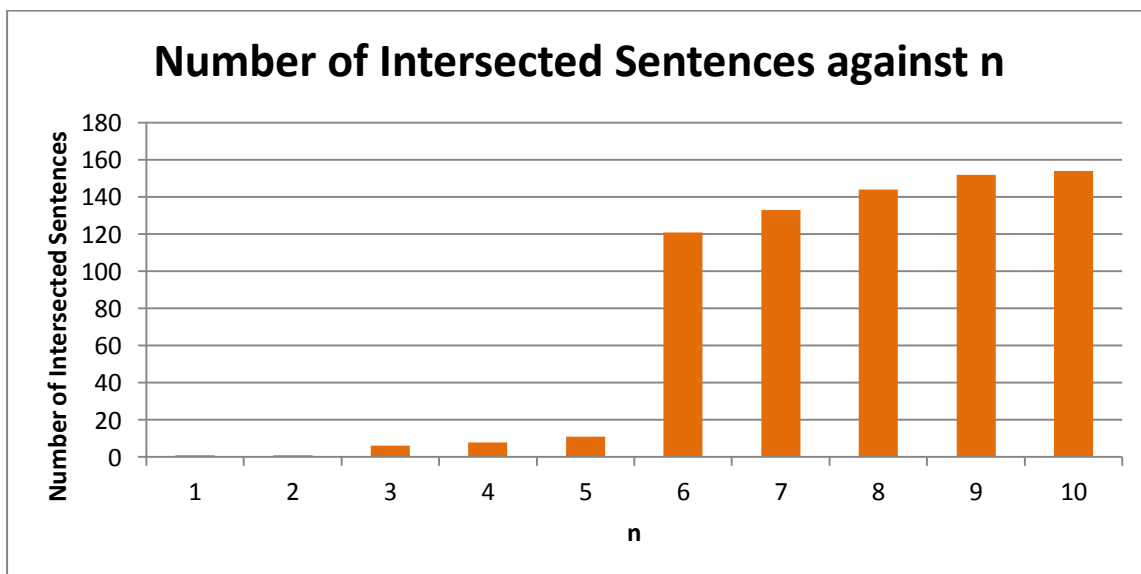


Figure 5-25: Graph of the Number of Intersected Sentences against n for Sentence 3

Sentence 4:

“A recall was issued for foil-wrapped Pirate's Gold chocolate coins because they contain melamine?”

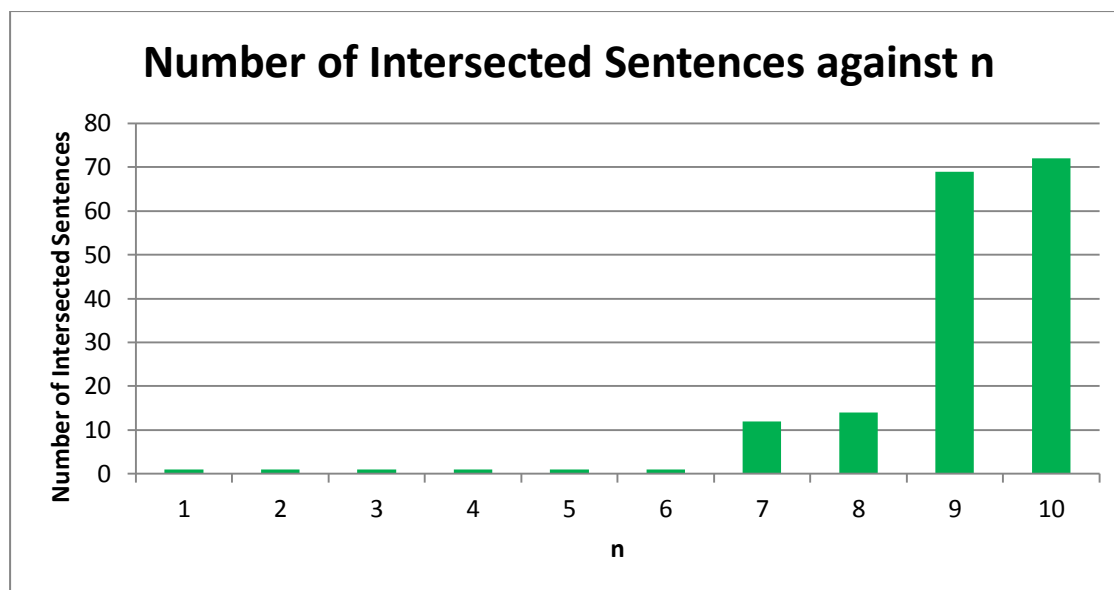


Figure 5-26: Graph of the Number of Intersected Sentences against n for Sentence 4

Sentence 5:

“Does a single dose of Children's Motrin cause ulcers and gastrointestinal bleeding in children?”

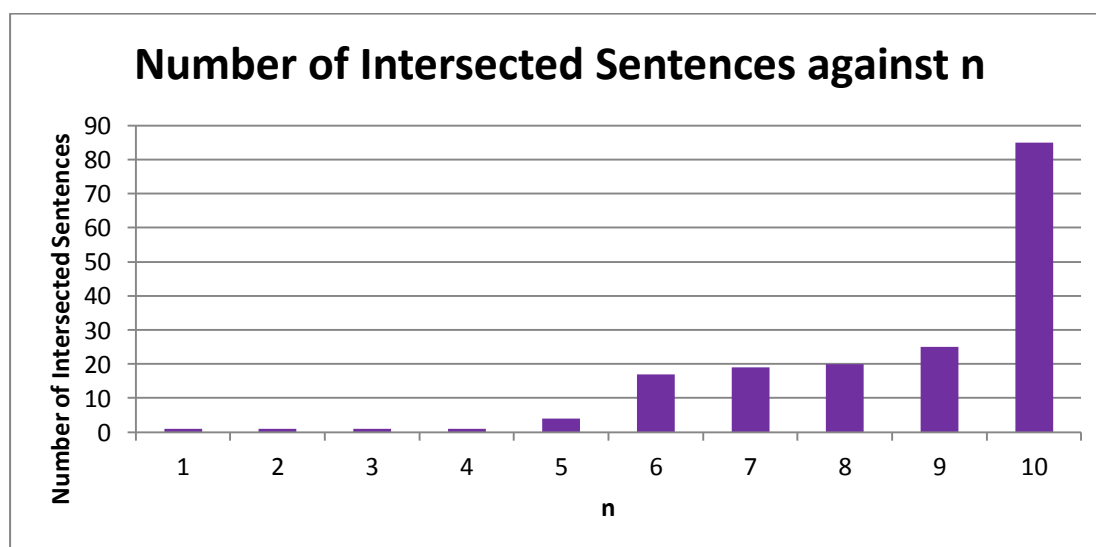


Figure 5-27: Graph of the Number of Intersected Sentences against n for Sentence 5

Therefore, based on the above graphs, it can be seen that when $n=5$, the number of retrieved sentences is between 5 to 11, which achieves the purpose of searching a suitable n value for the n -gram for query relaxation so that the number of sentences retrieved are not too many or too few.

The precision and recall rates were calculated based on the test sentences in Appendix D that was restructured from the original sentence from Doc_ID. The overall recall rate obtained is 89.8496833579735% with the overall precision rate being 80.0%. Based on the recall and precision rates given that the sentences were much shorter, words were in different order and using different terms as they were written using a human's understanding of the original sentence, the word order and word length does not affect the results obtained using the n-grams query method, but rather certain keywords especially health terms helped with the recall of the system.

The following table describes the test cases that are to be tested on the standalone system, where the first column is the item to be tested, the second and third column as the input and output which are the values given and returned by the system, and the expected output column for the expected output that should be seen and the result to state if the system is able to handle the test case by giving a "Pass" or "Fail".

Item to be tested	Input	Output	Expected output	Result
Link textbox is left empty	[empty string]	Error message: "Text Field is empty! Please enter a link for crawling."	Error message: "Text Field is empty! Please enter a link for crawling."	Pass
Invalid link is entered	abcd	Error message: "The text entered is not a valid URL! Please try again."	Error message: "The text entered is not a valid URL! Please try again."	Pass
Sentence for validation is empty	[empty string]	Error message: "Text Field is empty! Please enter a link for categorization."	Error message: "Text Field is empty! Please enter a link for categorization."	Pass
Sentence for validation returns no results	asda	Message: "There are no similar records in the database."	Message: "There are no similar records in the database."	Pass
Selects the option to save the sentence manually but the text field is empty	[empty string]	Error message: "No sentence was selected"	Error message: "No sentence was selected"	Pass
Selects the option to save highlighted sentence but no sentences were highlighted	[empty string]	Error message: "No sentence was selected"	Error message: "No sentence was selected"	Pass
Link to be crawled already exists in the database	http://www.hoax-slayer.com/elephant-hunting-family-photograph.shtml	Prompt message: "Link already exists in the database. Continue crawling?"	Prompt message: "Link already exists in the database. Continue crawling?"	Pass
The sentence to be saved is already in the database with the same web link	"Circulating protest message features an image of a dead elephant, still with food in its mouth, with a family of hunters posing behind the body."	Warning message: "This sentence for this webpage has already been saved."	Warning message: "This sentence for this webpage has already been saved."	Pass

Table 5-1: Black Box Testing Results for Standalone Java Application

CHAPTER 6: CONCLUSION

6.1 Project Review and Discussions

Hoaxes are articles/posts that can cause a person to shift their beliefs, and these cases are still very rampant especially with the rise of social media on the Internet. This poses a major problem as these hoaxes are mostly false, and therefore, may cause the reader to change his/her beliefs on false information.

To solve the above problem, a Google Chrome extension is developed that extracts the sentence that the user highlights and sends to the server to compute the semantic similarity between the highlighted sentence and the sentences that are stored in the database, which are the descriptions of webpages that contain information about hoax and non-hoaxes.

In the database preparation stage, the server crawls the selected websites (such as www.hoax-slayer.com, www.snopes.com and www.webmd.com) and stores information such as the title, description and link of the webpage. Lemmatization, POS tagging and a search for the word's synonyms is performed and appended to the database for future query.

The extension collects the highlighted sentence and sends to the server for categorization and retrieving related links to the sentence. POS tagging and lemmatization is done as a preprocessing step, and query is with bigrams from the highlighted sentence. After retrieving a list of related links, semantic similarity is done between the highlighted sentence and the list of related descriptions, and the top 3 related links are analyzed and sent back to the extension with the categorization of the sentence.

The tools used for the preprocessing stage include the Stanford CoreNLP, POS Tagger and the JAWS API. Path is the similarity measure used to calculate the semantic similarity between two words using WordNet. A complete Bipartite mapping is used to calculate the semantic similarity between a pair of words that are taken from each sentence, and the highest score is taken into consideration when calculating the overall semantic similarity of the sentences.

To conclude, the precision rate obtained is 80.0% with a recall rate of 89.8496833579735% using sentences as in Appendix D. Furthermore, two functional applications were produced: a standalone Java application and a Google Chrome extension

CHAPTER 6: CONCLUSION

that are able to calculate the semantic sentence similarity and categorize if a sentence is a hoax or not.

6.2 Project Constraints

The extension developed can only be used on the Google Chrome browser, and is not supported by the mobile versions of the browser. In addition, the scope of the project is limited to only health hoaxes, and therefore other hoaxes such as celebrity and giveaway hoaxes are not taken into consideration. Furthermore, the extension only takes in English as other languages are currently not supported. Moreover, the user has to evaluate by their own self if a sentence has the content that they want to analyze before manually highlighting it and click on the browser action icon of the extension.

6.3 Problems Encountered

Some problems were encountered during the development duration of the project. One of the major issues faced was that the ws4j API did not return the supposed outcome. It turns out the API had some internal issues whereby it did not loop through all the senses to obtain the highest score. The solution to that issue was posted on Stackoverflow.com (2013) and implemented in the code. Another problem was encountered during the crawling of websites to extract information to be stored in the database. Because each webpage is designed differently, it became a challenge to extract all the important text (such as the main article's content) to store in the database. Furthermore, there are many semantic similarity measures that have been proposed till today, and to select one that would fulfill the objectives of this project, multiple trial and error is done with different similarities before selecting the one that suits the purpose of this project.

6.4 Future Work and Enhancement

The developed extension and standalone application still has many flaws. To further enhance and improve the developed applications, it is suggested to collect each sentence in the main content of the hoax/non-hoax webpage so that a more detailed and higher recall and precision rate can be achieved as the description of the webpage is still not sufficient enough to determine a completely different sentence if it is a hoax or not. Furthermore, the application can only process English and therefore it is suggested to include other languages as well so that more users can benefit from the application. In addition, the application should

CHAPTER 6: CONCLUSION

widen its scope to include other categories of hoaxes and further categorize the hoax to its own category (e.g. health, celebrity, scam, etc).

One major improvement that the application should implement in the future is to store the links in the database similar to that of WordNet, by grouping words that have relation together as synsets. This is implemented as a query relaxation method, whereby each synset is similar to that of WordNet, with its own gloss and definition, but is custom made with only words from the stored words/sentences in the database. The search will start at the lowest common subsumer so that all related concepts will be included in the retrieved sentences, thereby increasing the recall rate.

BIBLIOGRAPHY

BIBLIOGRAPHY

Abbasi, J. 2011, *Is Trypophobia a Real Phobia?*. Available from:

<<http://www.popsoci.com/trypophobia>> [30 March 2015].

Achananuparp, P., Hu, X. and Shen, X. 2008, 'The Evaluation of Sentence Similarity

Measures', *Data Warehousing and Knowledge Discovery*, pp.305-316. Available from:

<http://scholar.google.com.my/scholar_url?url=http://www.researchgate.net/profile/Pala_korn_Achananuparp/publication/220802383_The_Evaluation_of_Sentence_Similarity_Measures/links/0deec52cb85c20b04a000000.pdf&hl=en&sa=X&scisig=AAGBfm0DPI5pZkMOxy4kx7_4jfCcEDSBVQ&nossl=1&oi=scholar&ei=HfgYVaj9N9C2uQSrqoHYDg&ved=0CBoQgAMoADAA> [30 March 2015].

Aliguliyev, R. 2009, 'A new sentence similarity measure and sentence based extractive technique for automatic text summarization', *Expert Systems with Applications*, vol. 36 no.4, pp.7764-7772. Available from:

<<http://www.sciencedirect.com/science/article/pii/S0957417408008737#>> [30 March 2015].

Amadei, M. 2015, *UCanAccess-A Pure Java JDBC Driver for Access*. Available from:

<<http://ucanaccess.sourceforge.net/site.html>> [2 April 2015].

Banerjee, S. and Pedersen, T. 2002, 'An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet', *Computational Linguistics and Intelligent Text Processing*, pp.136-145. Available from:

<<http://www.d.umn.edu/~tpederse/Pubs/cicling2002-b.pdf>> [1 April 2015].

Chuah, M. 2014, *Lecture 4 - Data Mining for Business Intelligence*.

Clark, P. 2010, Query Relaxation in AURA. Available from:

<http://www.cs.utexas.edu/users/pclark/working_notes/relaxation-wn41.pdf> [2 April 2015].

Code.google.com 2015, *crawler4j - Open Source Web Crawler for Java - Google Project Hosting*. Available from: <<https://code.google.com/p/crawler4j/>> [30 March 2015].

de Kok, D. and Brouwer, H. 2011, *Natural Language Processing for the Working*

Programmer, pp.25-29. Available from: <<http://www.nlpwp.org/nlpwp.pdf>> [1 April 2015].

BIBLIOGRAPHY

- Depraetere, I. and Langford, C. 2012, *Advanced English grammar*. London: Continuum International Pub.
- Developer.chrome.com n.d., *Overview - Google Chrome*. Available from: <<https://developer.chrome.com/extensions/overview>> [30 March 2015].
- Docs.oracle.com, n.d., *Windows System Requirements for JDK and JRE*. Available from: <https://docs.oracle.com/javase/8/docs/technotes/guides/install/windows_system_requirements.html#BABFBCEE> [30 March 2015].
- Evans, A., Martin, K. and Poatsy, M. 2010, *Technology in action*. Upper Saddle River, N.J.: Prentice Hall.
- Greenbacker, C. n.d., *WordNet Similarity Metrics*.
- Hoax-slayer.com 2014, *Fake Big W Facebook Pages Promise Large Prizes for Sharing*. Available from: <<http://www.hoax-slayer.com/big-w-factory-sealed-dell-computers-galaxy-scam.shtml>> [30 March 2015].
- Hoax-slayer.com 2014, *HOAX - 'Breast Larvae Infestation From Undergarments'*. Available from: <<http://www.hoax-slayer.com/breast-larvae.html>> [30 March 2015].
- Ideaeng.com 2014, *Difference Between Stemming and Lemmatization? - New Idea Engineering*. Available from: <<http://www.ideaeng.com/stemming-lemmatization-0601>> [31 March 2015].
- Javatechig 2012, *What is Rapid Application Development Model | JavaTechig*. Available from: <<http://javatechig.com/se-concepts/rapid-application-development-model>> [4 April 2015].
- Lee, M., Chang, J. and Hsieh, T. 2014, 'A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences', *The Scientific World Journal*, pp.1-17. Available from: <<http://www.hindawi.com/journals/tswj/2014/437162/>> [30 March 2015].
- Li, Y., McLean, D., Bandar, Z., O'Shea, J. and Crockett, K. 2006, 'Sentence similarity based on semantic nets and corpus statistics', *IEEE Transactions on Knowledge and Data Engineering*, vol.18, no.8, pp.1138-1150. Available from: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1644735>> [30 March 2015].

BIBLIOGRAPHY

Lin, D. 1998, 'An Information-Theoretic Definition of Similarity', *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, pp.296-304. Available from:

<https://www.google.com.my/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CCIQFjAA&url=http%3A%2F%2Fwebdocs.cs.ualberta.ca%2F~lindek%2Fpapers%2Fsim.pdf&ei=xXYfVc22BcfhuQTj7YHoBA&usg=AFQjCNFQcUf-ChyFzCu7PBvXI8pe8Dj_Xw&bvm=bv.89947451,d.c2E> [1 April 2015].

Ling.upenn.edu n.d., *Penn Treebank P.O.S. Tags*. Available from:

<http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html> [31 March 2015].

Ltcconline.net, n.d., *Mean, Mode, Median, and Standard Deviation*. Available from:

<<http://www.ltcconline.net/green/courses/201/descstat/mean.htm>> [2 April 2015].

Meng, L., Huang, R. and Gu, J. 2013, 'A Review of Semantic Similarity Measures in WordNet', *International Journal of Hybrid Information Technology*, vol. 6, no.1.

Available from: <http://www.sersc.org/journals/IJHIT/vol6_no1_2013/1.pdf>. [1 April 2015]

Miller, G. 1995, 'WordNet: a lexical database for English', *Commun. ACM*, vol.38, no.11, pp.39-41. Available from: <<http://dl.acm.org/citation.cfm?id=219748>> [31 March 2015].

Miller, G., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. 1993, *Introduction to WordNet: An On-line Lexical Database*. Available from:

<<http://wordnetcode.princeton.edu/5papers.pdf>> [31 March 2015].

Mozilla Developer Network 2015, *Sending and retrieving form data*. Available from:

<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms/Sending_and_retrieving_form_data> [30 March 2015].

Nlp.stanford.edu 2008, *Stemming and lemmatization*. Available from:

<<http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>> [31 March 2015].

Nlp.stanford.edu n.d., *The Stanford NLP (Natural Language Processing) Group*. Available from: <<http://nlp.stanford.edu/software/corenlp.shtml#About>> [30 March 2015].

BIBLIOGRAPHY

- Nlp.stanford.edu, n.d., *The Stanford NLP (Natural Language Processing) Group*. Available from: <<http://nlp.stanford.edu/software/tagger.shtml>> [30 March 2015].
- Pedersen, T., Patwardhan, S. and Michelizzi, J. 2004, 'WordNet::Similarity - Measuring The Relatedness Of Concepts'. Available from: <<http://www.d.umn.edu/~tpederse/Pubs/AAAI04PedersenT.pdf>> [1 April 2015].
- Pedersen, T., Patwardhan, S. and Michelizzi, J. n.d., *WordNet::Similarity::path - search.cpan.org*. Available from: <<http://search.cpan.org/dist/WordNet-Similarity/lib/WordNet/Similarity/path.pm>> [1 April 2015].
- Princeton University 2010, *About WordNet*. Available from: <<http://wordnet.princeton.edu>> [31 March 2015].
- Radford, B. 2014, *Social Media Ebola Hoax Causes Deaths : DNews*. Available from: <<http://news.discovery.com/human/psychology/social-media-ebola-hoax-causes-deaths-14100.htm>> [30 March 2015].
- Shima, H. n.d., *ws4j - WordNet Similarity for Java - Google Project Hosting*. Available from: <<https://code.google.com/p/ws4j/>> [1 April 2015].
- Slimani, T. 2013, 'Description and Evaluation of Semantic Similarity Measures Approaches', *International Journal of Computer Applications*, vol. 80, no. 10, pp.25-33. Available from: <<http://arxiv.org/ftp/arxiv/papers/1310/1310.8059.pdf>> [1 April 2015].
- Snopes.com 2013, *Oil of Olé*. Available from: <<http://www.snopes.com/medical/toxins/canola.asp>> [30 March 2015].
- Snopes.com 2014, *Malaysia Flight MH370 Video*. Available from: <<http://www.snopes.com/computer/facebook/malaysia.asp>> [30 March 2015].
- Snowball.tartarus.org n.d., *The Porter stemming algorithm*. Available from: <<http://snowball.tartarus.org/algorithms/porter/stemmer.html>> [31 March 2015].
- Softwaretestingfundamentals.com 2010, *Black Box Testing | Software Testing Fundamentals*. Available from: <<http://softwaretestingfundamentals.com/black-box-testing/>> [1 April 2015].

BIBLIOGRAPHY

- Softwaretestinghelp.com n.d., *Black Box Testing: Types and techniques of BBT — Software Testing Help*. Available from: <<http://www.softwaretestinghelp.com/black-box-testing/>> [1 April 2015].
- Song, W., Feng, M., Gu, N. and Wenyin, L. 2007, 'Question Similarity Calculation for FAQ Answering', *Third International Conference on Semantics, Knowledge and Grid (SKG 2007)*. Available from: <<http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.116.3776>> [30 March 2015].
- Spell, B. n.d., *Java API for WordNet Searching (JAWS)*. Available from: <<http://lyle.smu.edu/~tspell/jaws/>> [30 Mar. 2015].
- Stackoverflow.com 2013, *ws4j returns infinity for similarity measures that should return 1*. Available from: <<http://stackoverflow.com/questions/17750234/ws4j-returns-infinity-for-similarity-measures-that-should-return-1>> [1 April 2015].
- Support.google.com n.d., *Download and install Google Chrome - Chrome Help*. Available from: <<https://support.google.com/chrome/answer/95346?hl=en>> [30 March 2015].
- Tsonev, K. 2013, *Developing Google Chrome Extensions - Tuts+ Code Tutorial*. Available from: <<http://code.tutsplus.com/tutorials/developing-google-chrome-extensions--net-33076>> [30 March 2015].
- Vuković, M., Pripuzić, K. and Belani, H. 2009, 'An Intelligent Automatic Hoax Detection System', *Knowledge-Based and Intelligent Information and Engineering Systems*, pp.318-325. Available from: <http://www.ieee.hr/_download/repository/kes09.pdf> [30 March 2015].
- W3schools.com 2015, *Browser Statistics*. Available from: <http://www.w3schools.com/browsers/browsers_stats.asp> [30 March 2015].
- Web2.uvcs.uvic.ca n.d., *ELC Study Zone: Parts of Speech*. Available from: <<http://web2.uvcs.uvic.ca/elc/studyzone/330/grammar/parts.htm>> [31 March 2015].
- Weisstein, E. n.d., *Complete Bipartite Graph*. Available from: <<http://mathworld.wolfram.com/CompleteBipartiteGraph.html>> [1 April 2015].

BIBLIOGRAPHY

Wordnet.princeton.edu n.d., *WNGLOSS(7WN) manual page*. Available from:
<<https://wordnet.princeton.edu/man/wngloss.7WN.html>> [1 April 2015].

APPENDIX A: LIST OF POS TAGS USED IN THE PENN TREBANK PROJECT

The below shows a list of Part-Of-Speech Tags used in the Penn Treebank Project (Ling.upenn.edu, n.d.):

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential <i>there</i>
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRPS	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	<i>to</i>
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WPS	Possessive wh-pronoun
36.	WRB	Wh-adverb

APPENDIX B: SELECTION OF THE N VALUE FOR N-GRAM


Number of words in the sentence	Count	Cumulative Frequency
0	0	0
1	0	0
2	3	3
3	4	7
4	16	23
5	31	54
6	43	97
7	72	169
8	79	248
9	79	327
10	72	399
11	68	467
12	37	504
13	25	529
14	35	564
15	28	592
16	23	615
17	19	634
18	9	643
19	12	655
20	4	659
21	12	671
22	6	677
23	4	681
24	8	689
25	3	692
26	3	695
27	1	696
28	2	698
29	1	699
30	1	700
31	0	700
32	1	701
33	0	701
34	1	702
35	1	703
36	0	703
37	0	703
38	0	703
39	3	706
40	1	707

APPENDIX B

41	1	708
42	1	709
43	1	710
44	0	710
45	1	711
46	0	711
47	0	711
48	0	711
49	0	711
50	0	711
51	0	711
52	0	711
53	0	711
54	0	711
55	0	711
56	0	711
57	0	711
58	0	711
59	0	711
60	0	711
61	0	711
62	0	711
63	0	711
64	0	711
65	0	711
66	0	711
67	0	711
68	0	711
69	0	711
70	0	711
71	0	711
72	0	711
73	0	711
74	0	711
75	1	712
76	0	712
77	0	712
78	0	712
79	0	712
80	0	712
81	0	712
82	0	712
83	0	712
84	0	712

APPENDIX B

85	0	712
86	0	712
87	0	712
88	0	712
89	0	712
90	0	712
91	0	712
92	0	712
93	0	712
94	0	712
95	0	712
96	0	712
97	0	712
98	1	713
99	0	713
Total Number of Sentences	713	713

 Represents the term where the median lies

APPENDIX C: RESULTS FROM EXPERIMENTING WITH DIFFERENT VALUES OF N

Sentence	Message claims that a study found potentially harmful bacteria on lemons slices used in restaurants.
<i>n</i>	Number of intersected sentences
1	1
2	1
3	1
4	1
5	5
6	15
7	26
8	51
9	94
10	112

Sentence	Researchers found patients were more likely to be informed only after their disease had advanced
<i>n</i>	Number of intersected sentences
1	1
2	1
3	1
4	3
5	9
6	64
7	88
8	124
9	131
10	132

Sentence	Ankylosing spondylitis is a form of arthritis that is long - lasting (chronic) and most often affects the spine.
<i>n</i>	Number of intersected sentences
1	1
2	1
3	6
4	8
5	11
6	121
7	133
8	144
9	152
10	154

APPENDIX C

Sentence	A recall was issued for foil-wrapped Pirate's Gold chocolate coins because they contain melamine?
<i>n</i>	Number of intersected sentences
1	1
2	1
3	1
4	1
5	1
6	1
7	12
8	14
9	69
10	72

Sentence	Does a single dose of Children's Motrin cause ulcers and gastrointestinal bleeding in children?
<i>n</i>	Number of intersected sentences
1	1
2	1
3	1
4	1
5	4
6	17
7	19
8	20
9	25
10	85

APPENDIX D: PRECISION AND RECALL VALIDATION RESULTS

Doc_ID	New_Sentence	Precision	Recall
2	moving alert states that children's chocolate snack, Kinder Joy and styrofoam containers contains a wax coating that causes cancer.	1	1
4	notice states that staring at a BBQ fire when wearing contact lenses melts lenses and lead to eternal blindness.	1	1
9	Lengthy Internet notice states that pureed asparagus cures cancer and cites few cases as examples. The claim is that the information came from one Richard R. Vensal, D.D.S. and was printed in the 'Cancer News Journal' back in December 1979.	0	0
60	Illustration and feedback on Reusing Plastic Bottles causes cancer hoax	1	1
78	what is an anal abscess, causes, treatments and others	1	1
104	the tell-tale signs and types of attention deficit hyperactivity disorder.	1	1
110	treatment for adult ADHD	1	1
126	allergy reactions and symptoms range from mild to life-threatening.	1	1
142	more money spent on treatments, no difference in outcomes	1	1
143	the trend in the U.S. is not to give patients calming meds before procedure	1	1
152	degenerative disc disease is used to illustrate the normal changes in spinal disc due to age. Spinal discs are soft, compressible discs that separate the interlocking bones (vertebrae) which makes up the spine and acts as shock absorbers for the spine, allowing it to flex, bend, and twist.	1	1
191	causes of ear infections	1	1
204	symptoms and diagnosis of depression	0	0.312764635
224	the different causes of depression	0	0.609209994
231	depression treatments and methods of caring for depression symptoms with antidepressants with/without psychotherapy.	0	0.711397506
245	losing more weight can manage diabetes.	0	0.648188067
247	glucose is from foods that are made from carbohydrates and a blood glucose test measure this type of sugar.	1	1
259	benefits of cinnamon for diabetics	1	1
261	diabetic neuropathy is a complication of diabetes	1	1
271	soundwaves is able to estimate the position of objects in the surrounding area	1	1
288	vision tests measures ability to see the details at near and far distances, analyze for deficiencies in field of vision and ability to differentiate colors	1	1

APPENDIX D

303	benefits and different types of contact lenses	0	0.830008384
331	heart rate and debunks myths on a "too fast" or "too slow" heart	1	1
334	symptoms and causes of sciatica	0	0.856201931
350	causes, diagnosis and treatments of foot pain	1	1
355	advantages and disadvantages of various treatments such as medication, therapies, and mind-body techniques for pain	0	0
366	Effects and treatments for low testosterone	0	0.213029187
388	vision problems from drugs as side effect	1	1
420	side effects of oversleeping	0	0.744041975
431	return to sleep in the middle of the night and when to go for treatment	1	1
471	Reese's Peanut Butter Cups maggot infestation video	1	1
477	Kraft Food's Macaroni & Cheese Dinner recalled because of metal fragments contamination.	1	1
480	high levels of arsenic in california wine	1	1
489	cooking corn cobs in portable picnic coolers	1	1
517	store-bought extra virgin olive oils in U.S. are fake	1	1
527	dangerous cyanide compound found in apple seeds	1	1
559	complains about stale bread by a customer caused invention of french dip sandwich	1	1
565	tests to detect stroke	1	1
583	graviola tree fruit natural cancer cell killer	1	1
589	Kenyan tetanus vaccine campaign a secret sterilization movement	1	1
600	use flour for treatment of burns	1	1
602	soaking okra overnight and drinking the water cures diabetes	1	1
612	fast food companies say those overweight are more healthier	1	1
616	skin cancer not related to sun exposure	1	1
631	Cold-fX 'feeds' development of hormonal cancers in women?	1	1
653	patients who will donate their organs left to die by doctors	1	1
661	expectations from mayo clinic on the swine flu pandemic	1	1
663	how to avoid swine flu	1	1
665	Chinese astrological symbols predicts recent influenza outbreaks	1	1
700	recycled condoms used in chinese-made hair bands	1	1