1.0 INTRODUCTION

As explained in the first part of the project, there are numerous methods that have been proposed to be used for spell-checking tasks, which consists of error detection and error correction. Furthermore, as there are more methods available for error correction than error detection, the current project has decided to select a method than can be used in both error detection and correction for a more parsimonious system. Therefore, the spell-checking system of the current project will be based on dictionary lookup and n-gram analysis. In addition to that, the spell check of real word and non-word will be separated for lower complexities and lower runtime.

Hence, the report will be structured where the steps taken to obtain the dataset from a particular domain will be outlined. Then, the design, implementation, and evaluation of the spell-checking system will be discussed in detail. Following the discussion on the system architecture, the graphical user interface (GUI) and the elements in the GUI will be introduced. Finally, the system performance will be evaluated, and improvement techniques will also be proposed.

1.1 DOMAIN AND DATASET

The domain of the current project was chosen with the primary task-spell checking in mind so there must be sufficient amount of texts for model training and evaluation. Therefore, several scientific domains were considered, and the field of forensic psychology was chosen. The field of Psychology was chosen as it is a very language-driven field as the most direct way for individuals to express or explain their behaviours is through language communication. In other words, the theories, findings, or concepts are explained primarily in words. However, considering the fact that psychology is a broad field with different focus, the current project has decided to narrow the focus to forensic psychology.

After domain is chosen, research articles and books were considered as potential sources of datasets. As different sources were reviewed, textbooks about forensic psychology was deemed as the most appropriate source for dataset in comparison to journals and other online articles. The choice was decided based on the following factors: (1) The presentation of the texts and (2) The amount of information covered. As research articles have a much narrower focus, typically with only few research objectives, the amount of information covered in the field may not be sufficient. Additionally, the formatting of research articles may also vary following each publishers' formatting. Meanwhile, textbooks often have a set format

throughout the entire text, which makes the extraction of data much more efficient. Besides, textbooks are often written to cover wide variety of important topics in the field. Hence, textbooks are more likely to contain wider variety of information and contains higher amount of unique terminology of the field. In this case, a textbook on forensic psychology titled: “Handbook of forensic psychology” was chosen as the source of the dataset.

1.2 DATA SAMPLING AND PRE-PROCESSING

For the purpose of subsequent analyses, the dataset is first divided into 60% of training data, 20% of validation, and 20% of test data. As the textbook has different chapters that are discussing various different topics, simple separation of the corpus based on chapters cannot be used as a sampling method as this risks the model being unable to recognize the words in other chapters. The sampling technique is illustrated in the diagram below.

As shown in the diagram below, the textbook consists of 4 parts while each part contains several chapters. To ensure the model captured as many topics as possible, each chapter is further broken down into 60% of training dataset, 20% of validation set, and another 20% of testing dataset. After that, the 60% of training data extracted from each chapter are combined to form the training dataset. Then, all validation and testing datasets are also combined based on their respective functions.

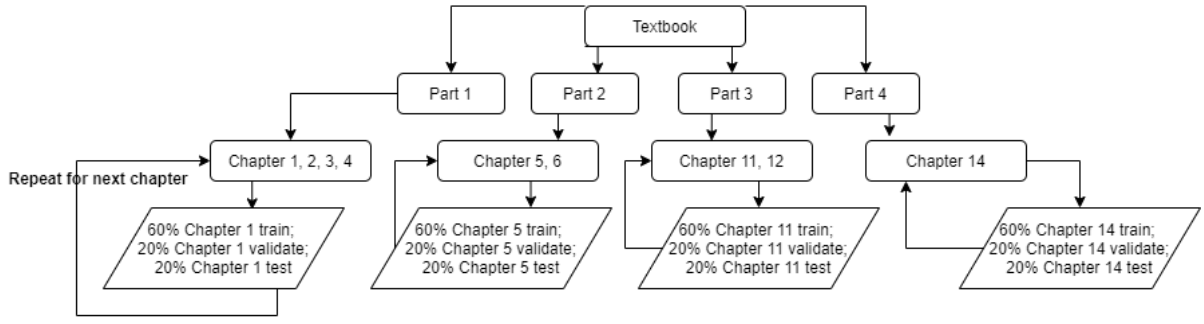
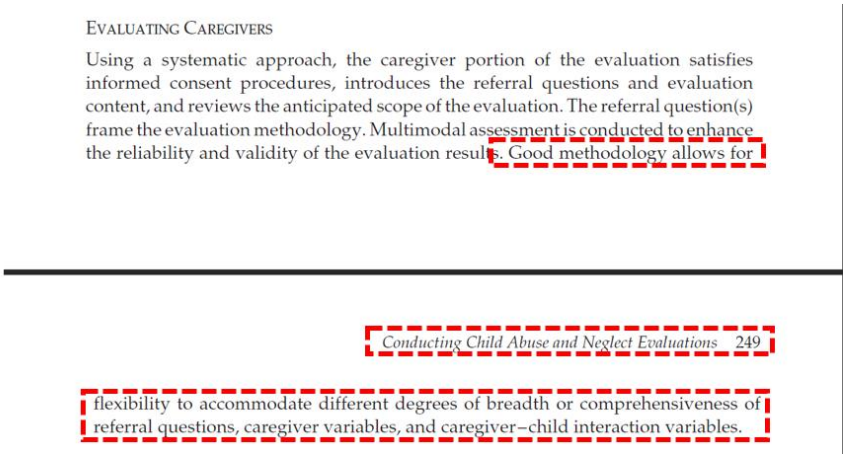


Figure 1.1 Data sampling - splitting data into 60/20/20.

After the sampling technique is confirmed, the format and presentation of the textbook is examined so that the subsequent steps required for data extraction and preparation can be planned. As the selected textbook is in pdf format, the file was first converted to a word document before any processing is conducted. During the inspection, it was found out that there are sentences that continue in the next page and page numbers that may pose potential issue to the extraction. As shown in the image above, some sentences may continue in the next page.

Directly extracting from the file may result in the subtitle and page number being placed between the sentence, which disrupts the overall meaning of the sentence. It may not pose any issue to a dictionary; however, the generation of bi grams may be difficult if the subtitles and page numbers are included during the extraction.


EVALUATING CAREGIVERS

Using a systematic approach, the caregiver portion of the evaluation satisfies informed consent procedures, introduces the referral questions and evaluation content, and reviews the anticipated scope of the evaluation. The referral question(s) frame the evaluation methodology. Multimodal assessment is conducted to enhance the reliability and validity of the evaluation results. Good methodology allows for

Conducting Child Abuse and Neglect Evaluations 249

flexibility to accommodate different degrees of breadth or comprehensiveness of referral questions, caregiver variables, and caregiver–child interaction variables.

Figure 1.2 Corpus with subtitles and page numbers

In order to avoid such issue, the corpus is first broken down to paragraphs and saved in a list. Subsequently, the extraction of text is performed by manually indexing the lists. By doing so, the subtitles and page numbers will not be included in the extraction and the continuity of the sentences can be ensured. The code below shows the manual indexing during data extraction.


```
In [8]: preface_text = get_complete_text(completedText, 118, 131) + get_complete_text(completedText, 134, 148) + get_complete_t
        chapter_1_text1 = 'I' + get_complete_text(completedText, 526, 540) + " " + get_complete_text(completedText, 543, 547) +
        chapter_1_text2  = get_complete_text(completedText, 560, 562) + " " + get_complete_text(completedText, 563, 565) + ". "
        chapter_1_text3  = get_complete_text(completedText, 596, 603) + " " + get_complete_text(completedText, 604, 607) + " "
        chapter_1_text4  = get_complete_text(completedText, 647, 648) + " " + get_complete_text(completedText, 650, 661) + " "
        chapter_1_text5  = get_complete_text(completedText, 686, 690) + " " + get_complete_text(completedText, 692, 696) + " "
        chapter_1_text6  = get_complete_text(completedText, 710, 714) + " " + get_complete_text(completedText, 718, 720) + " "
        chapter_1_text7  = get_complete_text(completedText, 754, 756) + " " + get_complete_text(completedText, 762, 772) + " "
        chapter_1_text8  = get_complete_text(completedText, 816, 820) + " " + get_complete_text(completedText, 822, 824) + " "
        chapter_1_text9  = get_complete_text(completedText, 838, 842) + " " + get_complete_text(completedText, 846, 848) + " "
        chapter_1_text10  = get_complete_text(completedText, 890, 892) + " " + get_complete_text(completedText, 894, 896) + " "
        chapter_1_text11 = get_complete_text(completedText, 918, 920) + " " + get_complete_text(completedText, 922, 926) + " "

In [9]: get_complete_text(completedText, 118, 131)
```

Figure 1.3 Corpus with subtitles and page numbers

After the extraction is completed, the dataset is processed through a series of procedures. The following flowchart illustrates the overall procedures involved.

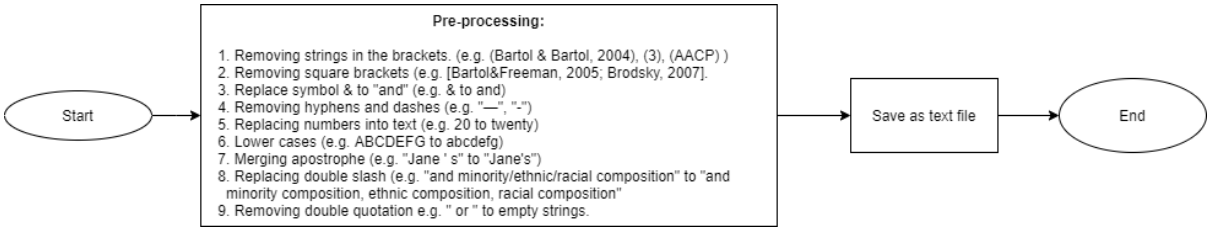


Figure 1.4 Series of procedures to do pre-processing

As shown in the flowchart above, the extracted dataset is saved as a text file. Any citations including the brackets are removed as the first pre-processing step. Then, symbols such as "&" are replaced to words "and". The text is then converted to lowercase with the dashes and hyphens removed. Subsequently, any abbreviations are also removed from the text and the text is then tokenized. Following tokenization, the tokenized file is processed again with numbers, apostrophe. Words that are separated by slash "/" are concatenated to form meaningful phrases. For instance, the sentence below shows multiple phrases separated. In order to overcome such issue, the symbol "/" is removed and words are concatenated to form phrases such as "minority composition", "ethnic composition", and "racial composition". After that, double quotations are also removed from the text as well. Finally, the processed file is saved as a text file again. Furthermore, the entire process is performed on each chapter of the textbook and saved in separate text files.

Disabilities Act of 1990, gender issues in policing, and minority/ethnic/racial composition of law enforcement agencies. Because this chapter focuses on early history,

Figure 1.5 The symbol "/" in the corpus

1.3 DICTIONARY OF WORDS

By using the pre-processed text files for each chapter, all the text files are loaded into one object. Subsequently, the combined texts are tokenized. Punctuations are also removed from the text as well. Then, the text file is converted to a dictionary that stores each individual word along with its' respective frequency distribution. The completed dictionary is saved using a JSON dump. The screenshot below shows an example of the dictionary.

```
In [28]: # To read json
         fd_1 = json.load(open("dictionary_freq_trainset.txt"))
         fd_2 = json.load(open("dictionary_freq_validationset.txt"))
         fd_3 = json.load(open("dictionary_freq_testset.txt"))

In [29]: fd_1.items()

Out[29]: dict_items([('the', 4081), ('potential', 36), ('for', 598), ('psychologists', 200), ('to', 1835), ('assist', 33),
         ('legal', 284), ('system', 89), ('has', 170), ('been', 169), ('recognized', 7), ('since', 22), ('early', 34), ('twe
         ntieth', 10), ('century', 16), ('but', 121), ('only', 71), ('within', 35), ('past', 24), ('fifty', 25), ('years', 3
         5), ('psychology', 176), ('begun', 3), ('realize', 1), ('this', 354), ('in', 1528), ('meaningful', 7), ('ways', 2
         5), ('progress', 8), ('included', 21), ('newly', 2), ('developed', 33), ('professional', 65), ('organizations', 1
         0), ('such', 151), ('as', 526), ('american', 32), ('law', 234), ('society', 12), ('and', 1952), ('international',
         4), ('association', 14), ('correctional', 32), ('forensic', 231), ('graduate', 4), ('internship', 5), ('fellowship
         ', 1), ('programs', 13), ('specialty', 28), ('area', 26), ('devoted', 9), ('certifying', 1), ('qualified', 7), ('pr
         actitioners', 26), ('board', 10), ('of', 2305), ('police', 53), ('public', 27), ('safety', 3), ('scientific', 18),
         ('journals', 11), ('human', 8), ('behavior', 41), ('behavioral', 11), ('sciences', 3), ('criminal', 92), ('justice
         ', 28), ('books', 13), ('interface', 2), ('continued', 12), ('grow', 1), ('rapidly', 2), ('previous', 12), ('editio
         n', 6), ('handbook', 8), ('was', 255), ('published', 28), ('two', 89), ('thousand', 24), ('six', 29), ('with', 42
         8), ('increasing', 4), ('numbers', 2), ('becoming', 10), ('involved', 29), ('practice', 58), ('research', 148), ('a
         ', 1163), ('steady', 2), ('flow', 2), ('new', 34), ('ideas', 4), ('information', 150), ('available', 39), ('chapter
         s', 2), ('nineteen', 117), ('twenty', 62), ('then', 36), ('provide', 85), ('accounts', 6), ('development', 19), ('l
         ie', 4), ('detection', 2), ('hypnosis', 1), ('describe', 14), ('current', 22), ('emerging', 2), ('trends', 5), ('us
         es', 3), ('these', 214), ('procedures', 31), ('part', 24), ('five', 43), ('looks', 2), ('at', 183), ('effective', 1
         2), ('communication', 7), ('expert', 118), ('opinion', 39), ('cases', 117), ('chapter', 34), ('one', 192), ('focuse
         s', 7), ('on', 373), ('essentials', 2), ('writing', 10), ('appropriate', 23), ('useful', 28), ('reports', 23), ('di
```

Figure 1.6 Screenshot of the dictionary

1.4 DICTIONARY OF BI-GRAMS

Similar to the steps above, the text files of each chapter are first loaded into one object. Then, the sentences are first segmented. Subsequently, using regular expression, bi-grams are created by zipping each pair of adjacent words together. Punctuations are also removed from the text and the completed dictionary of bi-grams is saved through a JSON dump.

```
In [28]: # Training set
         words_train_set = re.findall("\w+", train_set)
         bigram_freq_training_set = Counter(zip(words_train_set, islice(words_train_set, 1, None)))
         print(bigram_freq_training_set)

         Counter({('of', 'the'): 524, ('in', 'the'): 331, ('to', 'the'): 232, ('and', 'the'): 170, ('it', 'is'): 149, ('that
         ', 'the'): 146, ('on', 'the'): 121, ('by', 'the'): 116, ('to', 'be'): 114, ('in', 'a'): 108, ('for', 'the'): 103,
         ('the', 'plaintiff'): 97, ('with', 'the'): 94, ('mental', 'health'): 83, ('the', 'court'): 78, ('as', 'a'): 77, ('t
         he', 'defendant'): 75, ('the', 'legal'): 74, ('of', 'a'): 71, ('the', 'expert'): 71, ('the', 'case'): 68, ('has', '
         been'): 65, ('at', 'the'): 65, ('is', 'not'): 63, ('forensic', 'psychology'): 61, ('as', 'the'): 60, ('the', 'evalu
         ation'): 60, ('the', 'first'): 58, ('such', 'as'): 57, ('about', 'the'): 57, ('based', 'on'): 56, ('the', 'evaluato
         r'): 56, ('may', 'be'): 55, ('child', 'custody'): 55, ('in', 'which'): 53, ('in', 'this'): 51, ('of', 'their'): 51,
         ('in', 'nineteen'): 51, ('supreme', 'court'): 51, ('have', 'been'): 51, ('can', 'be'): 51, ('should', 'be'): 50, ('
         of', 'these'): 49, ('the', 'law'): 47, ('for', 'example'): 47, ('united', 'states'): 47, ('there', 'is'): 46, ('to
         ', 'a'): 46, ('not', 'be'): 46, ('is', 'a'): 46, ('as', 'well'): 45, ('health', 'professionals'): 44, ('risk', 'fac
         tors'): 44, ('the', 'psychologist'): 43, ('the', 'child'): 43, ('from', 'the'): 42, ('of', 'this'): 41, ('his', 'or
         '): 41, ('or', 'her'): 41, ('likely', 'to'): 40, ('of', 'violence'): 40, ('in', 'their'): 38, ('the', 'united'): 3
         8, ('risk', 'assessment'): 38, ('if', 'the'): 38, ('law', 'and'): 37, ('is', 'the'): 37, ('legal', 'system'): 36,
         ('related', 'to'): 36, ('number', 'of'): 36, ('the', 'same'): 35, ('well', 'as'): 35, ('must', 'be'): 35, ('the', '
         most'): 34, ('a', 'case'): 34, ('of', 'forensic'): 33, ('that', 'are'): 33, ('that', 'is'): 33, ('will', 'be'): 33,
         ('do', 'not'): 32, ('with', 'a'): 32, ('use', 'of'): 32, ('of', 'law'): 32, ('a', 'defendant'): 32, ('forensic', 'p
         sychologists'): 31, ('is', 'to'): 31, ('forensic', 'mental'): 31, ('to', 'provide'): 30, ('nature', 'of'): 30, ('th
         ey', 'are'): 30, ('is', 'important'): 30, ('violence', 'risk'): 30, ('the', 'attorney'): 30, ('does', 'not'): 29,
         ('would', 'be'): 29, ('one', 'of'): 28, ('that', 'a'): 28, ('for', 'a'): 28, ('in', 'addition'): 28, ('the', 'exami
```

Figure 1.7 Screenshot of Bi-gram dictionary

2.0 SYSTEM DESIGN

As the detection and correction methods for non-words and real words errors are separated. The discussion of the system design will be separated as well. However, the evaluation of both systems will be discussed in one single section.

2.1 SYSTEM DESIGN AND GUI

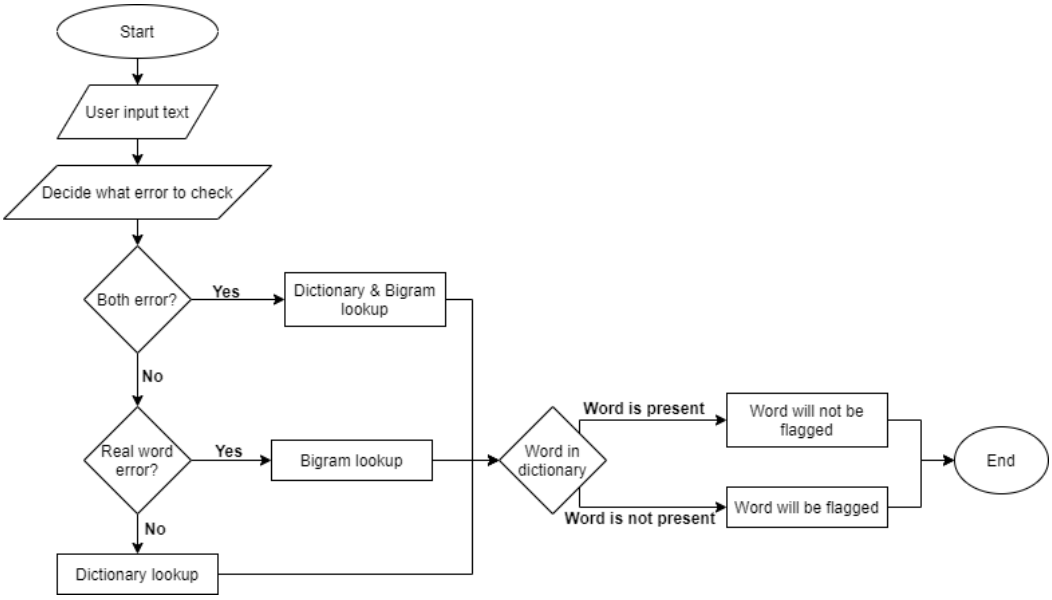


Figure 2.1 System flowchart

As the logic of determining the error will be discussed in detailed, the current section will focus on explaining the processes of the GUI and the overall procedures of the system following the input of the user. The procedure is displayed in the diagram above. The GUI will be displayed once the program is started, which is displayed in the screenshot below. The user must begin by inputting a body of text for checking.

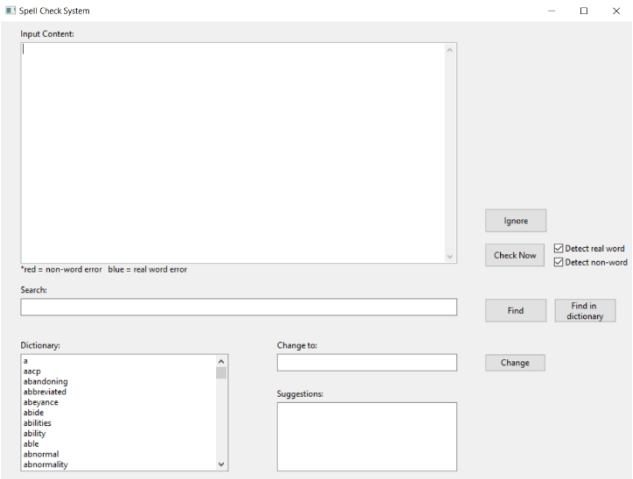


Figure 2.2 Screenshot of GUI

After inputting the text, the user can choose between 3 options, which are the checking for non-word error only, real word error only, or both types of errors at the same time. If non-word only is chosen, the spell-checker will only perform a simple dictionary lookup. If the word is present in the dictionary, the word will not be flagged. Otherwise, it will be flagged as a non-word error. Meanwhile, if real word only is chosen, the lookup will be performed in a bigram dictionary. If the word does not satisfy the rules that will be discussed in later sections, the word will be flagged. Finally, if both types of errors are chosen, both checking will be performed at the same time. Furthermore, in cases where both non-word and real word are detected in this option, the word will be searched in a dictionary first. If it is not present in the dictionary, the word will be flagged as a non-word error. However, if the word is present in the dictionary, it will be regarded as a real word error.

As both types of error may detect words that may not be necessarily error, another dictionary search option is provided in the GUI for user to search for flagged words in the dictionary. If the word is not present, an error message will be displayed to inform the user about the absence of the word. Following the detection of errors, the user can then highlight the flagged words and search for potential corrections that are generated based on edit distance and probability. Then, they can choose to ignore the error or replace one of the suggestions into the text.

2.2 NON-WORD ERRORS

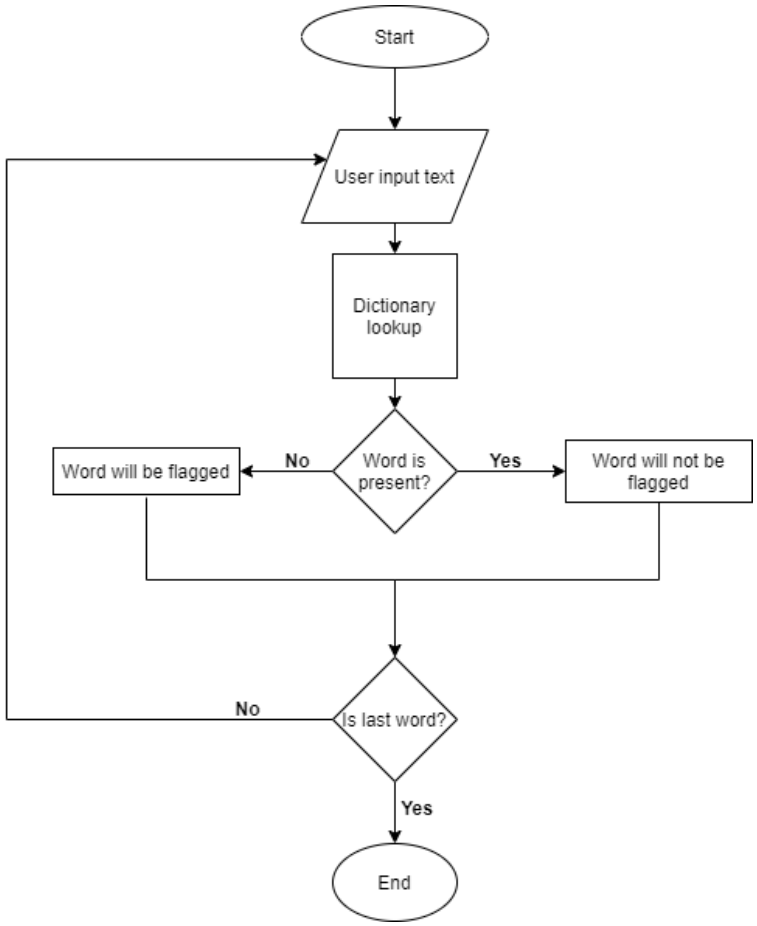


Figure 2.3 Flowchart diagram for non-word error detection

The checking of non-word errors is fairly straightforward, as outlined in the flowchart above. It is an iterative process where each word of the input text will be checked against the stored dictionary. If the word is present in the dictionary, the word will not be flagged as an error. Otherwise, it will be flagged as an error. The spell-checker process one word after another and the process will be in a iterative loop until all the words in the text are exhausted.

2.3 REAL WORD ERRORS

Although the text is also checked against a stored “dictionary” for real word error, the evaluation actually involves the representation of the relationship of all words in a corpus as a statistical model where each bigram has its’ respective probability. Additionally, the detection of real word errors is also different where a set of rules were incorporated into the process to ensure the context is adequately captured. The screenshot below shows the core of the logic in detecting real word errors.

```
# Approximates the probability of a word given all the previous words
# get bigram score from original content
# get bigram score from stemmed word
def check_realword_error(self):
    boolErrorList = []
    for note in self.bigram:
        content = note[0] + ' ' + note[1]
        stemmed_word = " ".join(token.lemma_.strip() for token in nlp(content)) # Stemming the word
        if ErrorDetection.get_bigram_score(content) == 0:
            # real word error
            boolErrorList.append(False)
        else:
            if ErrorDetection.get_bigram_score(stemmed_word) == 0 or ErrorDetection.get_bigram_score(stemmed_word) == ErrorDetection.get_bigram_score(content):
                boolErrorList.append(True)
            elif ErrorDetection.get_bigram_score(content) > ErrorDetection.get_bigram_score(stemmed_word) :
                boolErrorList.append(True)
            else:
                boolErrorList.append(False)

    self.text = boolErrorList
```

Figure 2.4 Screenshot of real word error logic

As mentioned, each bi-gram will possess a probability score which denotes the probability that the particular bi-gram occurs in the entire dictionary. Therefore, any pair of words that have occurred at least once in the dictionary will have a probability score of more than zero. Hence, the first rule of the current system determines that any phrase that has a probability of zero in a bi-gram dictionary will be deemed as a real-word error as the pair does not exist.

For phrases with probability more than zero, the system does not immediately regard them as correct words. A stemming method is employed for such cases where phrases will be stemmed to their root form and checked against the bi-gram dictionary. If the stemmed phrase returns a probability score of zero or is equal to the probability of the original phrase, the word will not be flagged. Meanwhile, if the original phrase probability score is higher than the stemmed phrase probability, it will not be flagged as well.

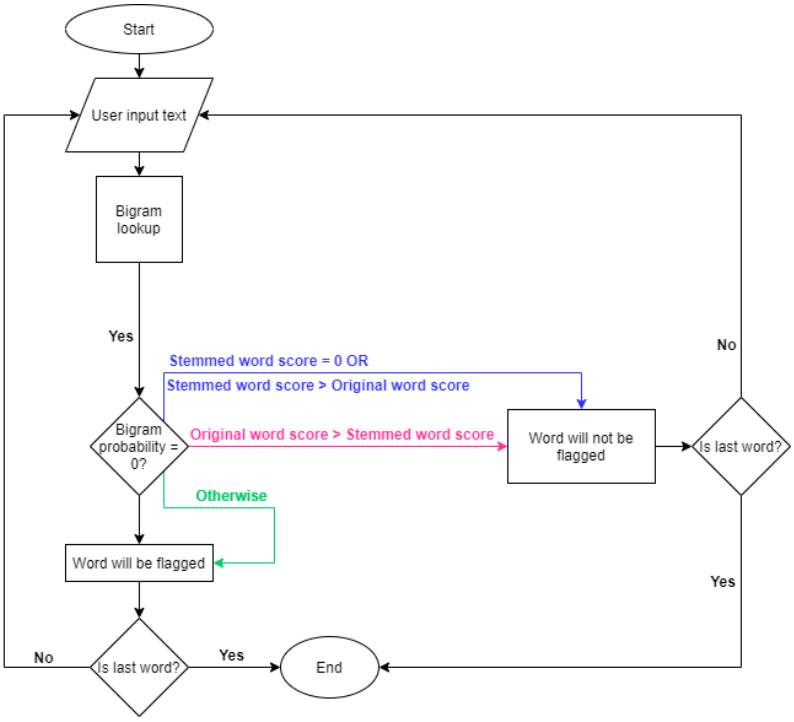


Figure 2.5 Flowchart diagram for real word error detection

3.0 MODEL EVALUATION

The model performance is measured through the calculation of precision and recall for both real-word and non-word errors detection. These metrics are chosen as they are widely used in many spell-checker evaluations studies (Van Zaanen and Van Huyssteen, 2003; Starlander and Popescu-Belis, 2002). Recall serves to indicate the model's sensitivity in detecting potential errors while precision indicates the accuracy of the model's detection. In other words, higher recall may indicate that the model is highly sensitive while high precision indicates that the detection made by the model is accurate. The calculation of recall and precision require few parameters, as shown in below formulae. X_1 is the total number of errors in the testing corpus while X_2 is the total number of detections. Finally, X_3 is total number of correct detections.

$$\text{Recall} = X_3 / X_1$$

$$\text{Precision} = X_3 / X_2$$

4.0 RESULT AND DISCUSSION

The table below summarizes the experiments conducted for the purpose of the evaluation.

Testers			Grant		Cheong		Chan		Choong	
Experiment	Dataset	Testing data	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)
			Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection
1	60% of corpus	20% of validation data with no synthetic errors	9	50	5	56	4	62	0	50
			Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)
			Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
1a	60% of corpus	20% of validation with 26 synthetic errors	100.00%	100.00%	87.50%	87.50%	90%	90%	70%	70%	100.00%	100.00%	100.00%	100.00%	88.89%	88.89%
			Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (real-word)	Results (real-word)
			Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection	Number of detection
2	60% of data + 20% of validation data	20% of testing data with no synthetic errors	12	48	6	53	8	47	8	48						
			Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)	Results (non-word)	Results (real-word)						
			Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
2a	60% of data + 20% of validation data	20% of testing data with synthetic errors (1 edit distance)	100%	77%	76.92%	84.62%	100%	91%	67%	100%	100%	100%	75.00%	100%	100%	100%	70%	100%
2b	60% of data + 20% of validation data	20% of testing data with synthetic errors (2 edit distance)	46.10%	100%	66.67%	66.67%	100%	100%	67%	100%	100%	100%	87.50%	100%	100%	100%	55.56%	100%

Figure 4.1 Result for different variation of testing

In order to ensure that the model is adequately trained, the validation corpus is first used to check the model performance where it was tested in two different experiments. In the first experiment, the validation corpus was submitted to the spell-checker without any modification to establish a baseline for the number of words that are unknown to the system. This step was taken as it is reasoned that the 60% of the dataset may not capture sufficient amount of words. By establishing the number of unrecognized words in the validation corpus, the calculation of precision and recall in the next experiment can be done more accurately and will not be confused with the synthetically introduced errors.

Following the establishment of the baseline, 26 errors (13 real word, 13 non-word) are introduced into the validation corpus for second testing. the detection for non-word are higher than detection of real words among all the testers. Furthermore, the results also suggest that the model was able to detect approximately 98% of the non-word errors and 98% of those detection were correct. Such performance was fairly good, which may also mean that the sampling technique used was effective in ensuring wide range of words from different chapters are captured in the dictionary. However, the performance for real word was much lower, with only 87% for both precision and recall.

Table 4.1 Results of experiment 1a (using validation data)

Average Recall (non-word)	Average Precision (non word)	Average recall (real word)	Average precision (real word)
97.50%	97.50%	86.60%	86.60%

Based on the model performance, the system is deemed complete and ready for the subsequent testing. Therefore, the validation set is then processed to tokens and bi-grams and subsequently added into the dictionary. Similar to the steps above, the remaining 20% of the data is used to test the re-trained model that contains the initial 60% of training data and 20% of validation data. The table below summarizes the results of the testing using the remaining 20% of unseen data. Using the corpus, errors with 1 and 2 edit distance were introduced and performance of the model on different edit distance is evaluated.

Table 4.2 Results of experiment 2a and 2b (using testing data)

Edit distance	Average recall (non word)	Average precision (non word)	Average recall (real word)	Average precision (real word)
1	100%	92%	72%	96%
2	87%	100%	69%	92%

Based on the results above, it can be seen that the overall performance of the detection of non-words is still better than the performance of real word detection. This is expected as the detection of non-word errors rely purely on the words' presence in the stored dictionary. On the other hand, the detection of real word error is less straightforward. As shown in the results, the model was only able to detect 60 to 70% of the real word errors. This may suggest that the model's sensitivity to errors and threshold may need further adjustment. Nonetheless, majority of the detection that were made by the model are accurate, as the precision is approximately 90% for both types of edit distance.

5.0 LIMITATIONS AND FUTURE IMPLICATIONS

Upon the implementation of the model, a limitation is noticed, which is the model's inability in detecting errors that occur at the first word of the text. In this case, the way to overcome such issue is the addition of an empty space before and after each sentence in the corpus. By doing so, it will inform the model regarding the words often used for the beginning and the end of each sentence.

In addition to that, the n-gram approach of the current project can also be further improved. Although n-gram is language agnostic as it conducts spell-checking by computing similarity, the order of the n-grams is often not considered. In the current project, an assumption was made whereby all the errors occur at the second word of any bi-gram pair. However, in real life, the error may occur in any position. In this case, the position of the bi grams should be considered for the spell-checker to model real life errors.

Besides that, the current system can also be further enhanced in terms of the ranking of the suggestions. As the current system ranks the suggestions based purely on edit distance and probability score. In this case, the incorporation of parts of speech (POS) tagging may help to make the ranking more accurate. As POS can recognize the roles played by each word in a sentence, such contextual information about the corpus can complement the edit distance and probabilistic methods.

REFERENCES

Starlander, M. and Popescu-Belis, A., 2002, May. Corpus-based Evaluation of a French Spelling and Grammar Checker. In Third International Conference on Language Resources and Evaluation, Canary Island, 29th to 31st May.

Van Zaanen, M. and Van Huyssteen, G. (2003). Improving a spelling checker for Afrikaans. *In Computational Linguistics in the Netherlands 2002*, Groningen, 29th November.