**School of Information Systems**
**Curtin Business School**

# A Robust Methodology for Automated Essay Grading

**Anhar Fazal**

**This thesis is presented for the Degree of**
**Master of Philosophy**
**of**
**Curtin University**

**April 2013**

# DECLARATION

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: …….

Date: ……...

# Acknowledgements

Firstly, I would like to thank Almighty God for blessing me with the ability, physical strength and motivation to complete my Masters by Research thesis under the supervision of Professor Tharam Dillon and Professor Elizabeth Chang.

Secondly, I wish to acknowledge the efforts of my family without whose support and sacrifices I would never have been the person that I am today. To my parents, Mohammed Khaja Moinuddin and Jafari Sultana, who have guided me and instilled the value of education early in life. To my lovely husband Dr. Farookh Hussain who believed in my ability to pursue research and has been my pillar of support through thick and thin since day one of my study. To my wonderful, loving and wise parents-in-law Dr. Khadeer Kaleemuddin and Mrs.Touqeer Khadeer, who have taught me a lot in gentle ways and bore with me patiently, I can never ever thank you enough for all that you have done for me. To my beautiful children Faadil and Farhaan, for giving me the reason to keep going despite hurdles in my way. To my brother-in-law Omar for all his help whenever I needed it and my sweet sister-in-law Salwa, I love the way we studied together late at night and broke into laughter discussing things. Lastly, to my lovely siblings and my sweet and affectionate niece, you add a lovely dimension to my life. To all my family, I hope that this thesis is a sign of the good things to come in life.

Then, I would like to thank my primary supervisor Professor Tharam Dillon and my co-supervisor Professor Elizabeth Chang for their never-ending support, never-ending compassion, encouragement and superb guidance. This thesis is as much as their effort as it is mine. Finally I would like to thank all the members in Digital Ecosystems and Business Intelligence (DEBI) Institute, for their help and support during the past three years.

# List of Publications from this thesis

(1) Anhar Fazal, Farookh Khadeer Hussain and Tharam Dillon, “An intelligent approach for automatically grading spelling in essays using rubric-based scoring”, Journal of Computer and System Sciences, DOI: 10.1016/j.jcss.2013.01.021

(2) Anhar Fazal, Tharam S. Dillon, Elizabeth Chang: Noise Reduction in Essay Datasets for Automated Essay Grading. Proceedings of the On the Move to Meaningful Internet Systems (OTM 2011) - Confederated International Workshops and Posters: EI2N+NSF ICE, ICSP+INBAST, ISDE, ORM, OTMA, SWWS+MONET+SeDeS, and VADER 2011, Hersonissos, Crete, Greece, October 17-21, 2011, pp. 484-493

# Abstract

Essays have been conventionally marked by humans using two types of methods – Holistic and Analytic. In Australia, analytic scoring method is used in the National Assessment Program – Literacy and Numeracy (NAPLAN). Due to the subjective nature of the essays, the task becomes burdensome over a period of time. Moreover, humans can be bogged by halo effects, leniency and fatigue, if they have been marking essays for a long time. Hence to overcome the limitations of the human marker, an automated essay grading (AEG) system is desirable.

Most of the available AEG systems can be used to perform holistic scoring. However, only a small number of them can be used for analytic scoring and none can be used to grade essays according to the NAPLAN rubric. This thesis is a humble effort to address this limitation. Specifically the objective of this thesis is to propose and develop a robust methodology for automated essay grading for grading essays based on the NAPLAN rubric.

To achieve the objective, several methodologies have been proposed and implemented for automatically grading spelling, vocabulary and sentence structure of the essays. Heuristics and rules based on English language and neural network modelling have been used to design and develop the methodologies. Each of them has been tested using a real world dataset. Furthermore, the results have been evaluated using precision, recall, F-measure and rater agreement metrics which are widely used in the field of data mining. Hypothesis testing and student t-tests are performed and it has been shown that there is insignificant difference between the scores assigned by human marker and this system.

# List of Figures

# List of Tables

# Table of Contents

# Chapter 1: Introduction

## 1.1. Introduction

In this chapter, the various types of assessments carried out across the world at various levels of education are discussed. The definition of an essay and the concept of automated essay grading (AEG) are provided. The two types of essay scoring systems prevalent in the educational system, holistic scoring and analytic scoring, are outlined. The various applications of AEG systems and the criticisms of these systems are presented. Although there is some criticism of AEG systems, the reality is that these systems now have a widespread use both in small-scale and large-scale assessments. This can be attributed to the manifold advantages of AEG systems over manual grading. The origin and background of AEG systems is detailed in this chapter, after which the aim, scope and most importantly, the significance of this thesis is discussed. To assist the readers locate a particular chapter of choice, a plan of this thesis in the form of a map is provided in the later part of this chapter. Finally, the chapter concludes with a summary of the points presented in each section.

In the next section, the different types of assessment questions used to test the ability of students are outlined.

## 1.2. Types of Assessment Questions

Across the world, at every level of education, whether primary, secondary or tertiary, it is an accepted norm that students' abilities will be tested by means of exams. Further, it is general practice in every country that when a student is promoted from one year level to the next year level, they need to demonstrate that they meet the minimum requirements to qualify for this progression. Consequently, a student's level of success or failure in the exams will determine if the student has the required level of capability to be promoted to the next year level. To assess students' abilities at various levels of education, typically, there are different types of questions in an exam. These include:

- Objective-type questions - These are also called 'Fill in the Blanks' where questions are framed in such a way that the answer must fit into the blank provided.

  For example: To be respectful is to treat someone with _____.

  Answer: respect.

- Multiple Choice Questions (MCQs) - These types of questions are used frequently in formative assessments. The question includes several possible answers, of which only one is correct and the student must choose the correct one.

  For example: To be respectful is to treat someone with

  a) humour b) hostility c) liveliness e) respect

  Answer: 'e'.

- Choose 'True' or 'False' - These questions are framed as a statement and the student must decide whether it is 'True' or 'False'.

For example: To be respectful is to treat someone with respect. (True/False)

Answer: True.

- Short answer-type questions - These questions are framed in such a way that the length of the answer will be between one word and 4-5 sentences.

  For example: What are the various colours in a rainbow?

- Comprehension-type questions- These questions are usually found at the end of a reading comprehension passage. The answers to these questions are in the reading passage and are used to check the student's ability to comprehend the text.
- Essay-type questions- These are the most interesting type of questions in the context of this thesis. An essay question can be either broad and open or narrow and specific. The answer should comprise several sentences in response to the question.

Conventionally, assessment by human markers has been the method for grading responses to all types of questions. However, in recent times, machine-based systems such as semi-automated and fully automated assessment systems have been developed and are being continuously improved to perform the same task as human markers. Machine-based systems that are used to automatically assess responses to the first three types of questions mentioned above are very easy to develop because they do not require a high level of sophistication. However, due to the subjective nature of responses for the last three types of questions mentioned above, machine-based systems developed to grade them can be considered at the highest level of the hierarchy. Furthermore, the automated assessment of short answer-type questions is easier compared to the automated assessment of essays. This point is supported by figure 1.1 which shows the increase in the marking complexity depending on the type of question being assessed. Figure 1.1 shows that open-ended questions such as creative essays, in other words narrative essays, are the most complicated for marking automatically. In this

thesis, the main focus is on developing a robust methodology of an automated essay grading system for marking narrative essays.

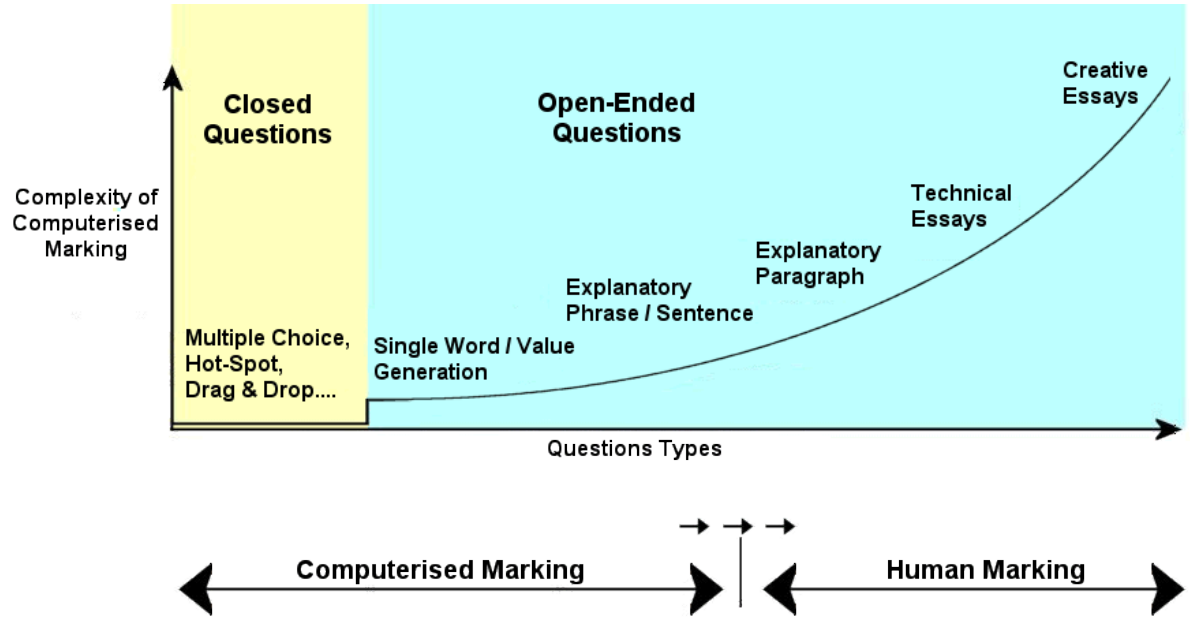



**Figure 1.1: Complexity of marking versus Question types (adapted from [1])**

In the next section, the formal definitions of essay and automated essay grading are provided.

## 1.3. Definitions of Essay and Automated Essay Grading

The Standard American English Dictionary defines an essay as "a written answer that includes information and discussion, usually to test how well the student understands the subject". An essay is a literary representation of a sequence of thoughts, mostly on a particular topic. The method of writing an essay while conforming to English language conventions

such as using proper spelling and grammar, and the ability to display coherence in the essay is a positive approach to writing an essay.

Automated Essay Grading (AEG) is defined as computer technology that evaluates and scores written prose [1, 2]. It is also referred to as computerized essay scoring, computer essay grading, computer-assisted writing assessment, Computer-based Assessment Methods (CbAS), Machine Scoring of Essays (MSE), Automated Essay Scoring systems (AES), Automated Writing Evaluation (AWE) and Computer-based Essay Marking System (CBEM). Regardless of the term used, these systems are primarily concerned with "the ability of computer technology to evaluate and score written prose" ([2], p.xiii). In this thesis, one or more of these terms is used inter-changeably, but most often the term AEG is used to denote an automated essay grading system.

The next section outlines the various methods for scoring an essay. Their methodology is discussed and examples are provided.

## 1.4. Types of Essay Scoring

Ellis Batten Page, the first developer of a fully automated essay grading system, advocated that "*content* loosely refers to what the essay says and *style* refers to syntax and mechanics and diction and other aspects of the way it is said" ([3], p:240). Further, content is 'what' the essay is about and style is 'how' the essay conveys its message. So ideally, for essay scoring, all the aspects of content and style are to be taken into account. The present methods of essay

scoring, while considering the aspects of content and style, can be classified into two: holistic scoring and rubric-based scoring.

### 1.4.1. Holistic scoring

Holistic scoring is also called global or impressionistic scoring [4]. In holistic scoring, the essay is assessed and a single score selected from a predefined score range is assigned as the overall score [5]. For example, the Graduate Record Examination (GRE) uses a score range of 0-6 for scoring essays, where score '0' is for an extremely poor essay and score '6' is for an exceptionally good essay [6].

It is important to note that the holistic score reflects the overall impression of the grader rather than the sum of the scores of individual features. This method is conventionally used because it does not require as much time as multi-trait scoring. More grading time would incur higher costs to complete the grading task [7]. Hence, this is the most common method of scoring and is even used in large-scale assessments such as GRE, GMAT and TOEFL. However, the drawback of this method is that specific strengths and weaknesses of students are not captured and furthermore, feedback reflecting the specific improvements required cannot be given comprehensively.

### 1.4.2. Multi-trait scoring

This method is also called the Analytical or Rubric-based scoring method [4]. In this method, essays are assessed based on a certain set of well-defined rubrics or features [5]. Each rubric has a scale associated with it and each score point is diligently explained. The final score awarded to the essay is the sum of scores of all the essay rubrics/features. For example, the National Assessment Program-Literacy and Numeracy (NAPLAN) Writing Assessment incorporates 10 essay features (such as spelling, vocabulary, ideas, sentence structure and so on) and each feature is associated with a scale (such as a scale of 0 to 6 for spelling, a scale of

0 to 5 for vocabulary) [8]. The advantages of this method are that it captures the specific strengths and weaknesses of an essay. Since the assessment of the essay is done based on well-defined rubrics, feedback can be provided to the student in great individual and specific detail.

In the next section, the need for an AEG system and the intended use of such systems is discussed.

## 1.5. Need for an AEG system

Grading essays is an essential part of a teacher's job. With the ever-increasing number of students, the number of essays to be graded also increases and it becomes monotonous and cumbersome for a teacher/marker to score the same task repeatedly. Hence, an automatic essay grader which takes a student's essay as an input and assigns a suitable grade as output is a great benefit to teachers. Many attempts have been made to realize such a system.

- **For writing assessment**. The primary use of AEG systems is for writing assessment, also called summative assessment. AEG systems are designed and developed for the purpose of low-stakes classroom assessment and large-scale high-stakes assessment (for example, national level examinations). The idea behind this technology is to reduce the teachers' workload, thereby freeing up some of the teacher's valuable time; but not to replace them [9].

- **For instructional purpose**. AEG systems are also used as an instructional tool for students by enabling the provision of both the essay score and feedback on the students' essays. This is also called formative assessment. AEG systems point out the areas within the essay that need further improvement on which the students can focus, while writing revised drafts of essays.

Keeping in view the various uses of AEG systems, the first AEG system was developed in the 1960s. Since then, there has been evolution in technologies due to the digital revolution in the 1990s and more and more AEG systems have been developed. The next section highlights this evolution of AEG systems from 1960s to present.

## 1.6. Background of the field of AEG systems

Although the earliest AES system made its debut in the 1960s, it was only in the 1990s that there was an upsurge in the number of AES systems being developed. This can be attributed to the fact that the micro- computer revolution in the 1980s made the machine assessment of written responses plausible [10].

The first effort in developing a computer-based essay grader was made by the late Ellis Batten Page, in 1966. He can be regarded as a pioneer in the field of AEG systems. Although Project Essay Grade (PEG) was released in 1966, the groundwork for this system had begun some years before [3] . Page built an automatic essay grader, and called it the Project Essay Grade. He successfully demonstrated that an automated "rater" is indistinguishable from human raters [11]. Furthermore, several studies reported that students "trusted" that a machine

could reliably mark their writing and were enthusiastic about writing online and receiving instant feedback [12].

During the 1990s, several new techniques, including natural language processing techniques, content analysis and artificial intelligence started emerging. This encouraged researchers to pursue the idea of developing AEG systems and hence approaches such as the Intelligent Essay Assessor (IEA), E-Rater, Educational Testing Service 1 (ETS1), Text Categorization Techniques (TCT) and Conceptual-Rater (C-Rater) emerged. Later, on there were other approaches such as Bayesian Essay Testing Scoring sYstem (BETSY), Intelligent Essay Marking System (IEMS); Schema, Extract, Analyse and Report (SEAR); Paperless School free text Marking Engine (PS-ME) and Automark. The most recent essay scoring methods are Intellimetric, MarkIT and Automatic Essay Assessor (AEA). Of all these methods, C-rater is a system for the automated marking/grading of short answer responses, which is an equally challenging task.

Over the past years, there has been an increasing interest and acceptance of AEG systems. This is evident from the number of references in the academic media publications [1].

Regardless of the number of AEG approaches, the basic idea is more or less, the same. A substantially large set of prompt-specific essays are pre-scored by human expert graders. This set is divided into two: a- training set and a testing set. The training set is used for developing the scoring model and attuning it. Then the scoring model is used to assign scores to the essays in the testing set. Most often, the benchmark for an AEG system is a set of essays with human-expert assigned scores. The AEG system is tested for its accuracy and efficiency in providing scores as close to the human- assigned scores as possible. The performance of the scoring model is typically validated by calculating how well the scoring model “replicated”

the scores assigned by the human expert graders. Hence, an AEG system that can produce grades at least as accurate as humans would be deemed usable [13].

In the next section, advantages of automated essay grading systems over manual grading are discussed.

## 1.7. Advantages of automated grading

AEG systems offer the following advantages over conventional methods of manual grading.

**Reproducibility**. An AEG system is built based on an algorithm. Hence it is capable of producing the same result if an essay is graded twice. This is not usually possible in human grading because of the variability in human judgement.

**Accuracy and Consistency**. Humans can be bogged down by halo effects, leniency and fatigue, if they have been marking essays for an extended period of time. Since an AEG system is free of these shortcomings, it will produce scores accurately and consistently over a period of time.

**Tractability**. An AEG system applies certain rules during the marking process. So, for every score assigned by the system, we can backtrack and find the source of any erroneous discrepancy. This cannot be done in human assessment.

**Cost efficiency**. Training an AEG system involves significantly less cost as opposed to training a few hundred human markers for large-scale assessment. After completing the training

phase of the AEG system, when it begins grading essays, it will result in further cost-savings, as minimal human effort will be required in grading at this stage.

**Adaptive testing**. Large-scale assessments can be made adaptive so that they suit the test-taker's ability. In an adaptive test like the GRE, the next question that is assigned to the test-taker depends on his response to the current question [14]. If the test-taker's answer to the current question is correct, then he is assigned a harder question. Otherwise, he is assigned an easier question.

**Time efficiency**. Essay grading is a time-consuming activity. By using AEG systems, hundreds of essays can be graded in a few minutes. In fact, some online AES systems boast that they can provide immediate scores and feedback. Hence [1, 12] emphasise that AEG systems can help achieve time efficiency.

**Re-usability**. The same AEG system can be re-trained and re-used if the scoring rubric changes [15]. The re-training of the AEG system would require nominal time and effort when compared to the re-training of hundreds of human markers.

**Better feedback**. From the student's viewpoint, obtaining feedback regarding written work is very important [16-18]. In fact, immediate and individual feedback has been proven to be very motivational and inspires the students to write better and to write more [12]. However, providing detailed feedback requires a lot of time. For this reason, most teachers cannot do this, when, at the same time they have to mark hundreds of thousands of essays. Several researchers report that AEG systems have the capability to provide detailed, individual feedback in a few seconds after the written work is submitted to the system [1, 12]. Some online AEG systems like Criterion and My! Access are already being used in educational institutions to help students write better, by improving on their written work according to the feedback received by students.

**No sequential rating effects**. Human graders have been found to suffer from sequential effects during essay grading. This means that an essay at hand is graded depending on the quality of the previous essay. Notably from [19], "a C paper may be graded B if it is read after an illiterate theme, but if it follows an A paper, if such can be found, it seems to be of D calibre"(p.41). AEG systems would not suffer from this kind of effect, primarily because every time a computer program is run, it applies the same set of rules to analyse an essay, irrespective of the quality of the previous essay that was graded.

**Other human-related factors that affect grading**. Several other human-related factors that affect essay grading performance has been widely researched over the past years [20]. Substantial evidence points to the conclusion that even though human graders have been rigorously trained, differences in their background and training and their experience in grading produce subtle but significant differences in the grading performance. Again, an automated program would be free of these shortcomings and its grading performance will not be affected by these factors.

Despite the many advantages of AEG systems as discussed above, a fraction of the academic community holds criticism against AEG systems. This is discussed further in the next section.

## 1.8. Criticism against AEG systems

Critics from the academic community express concerns over the use and effectiveness of AEG systems in educational settings [18]. They insist that a computer cannot "read" the text

and understand it in the way humans do, because of the fact that a computer is a machine. Additionally, critics claim that if students access AWE technology, then they might write text by using only the surface features or according to some formulaic pattern to try to "beat the system" and trick it into giving them a higher score [21]. Further, they argue that students might focus on writing a lengthy essay without giving much importance to the content of the essay. Moreover, students might under-estimate the importance of writing as a device of expression, because they are writing for a machine rather than for a human to read.

In contrast to these critical claims, most AEG systems demonstrate a correlation coefficient with human expert markers in the range of 75%-97%. Several AEG systems claim that their performance is at least as accurate as human expert graders and that their score agreement with a human is as much as the score agreement between two human expert graders [1, 2, 3, 4, 9].

There is considerable resistance in the academic community to let the computers be the sole judge in grading an essay. On the other hand, it is fairly well accepted that the computers play an auxiliary role in grading essays. For this reason, in large scale assessments like the GRE, an AEG system is used only in conjunction with a human grader. A second human grader is called in when a discrepancy arises between the AEG system and the human grader [22]. In fact, research studies demonstrate that when used in conjunction with a teacher, AES technologies pose huge advantages both for the teacher as well as for students [18]. Many researchers have questioned the validity and reliability of AEG systems. This sparked a number of validity and reliability tests that were carried out to see how valid and reliable the scores assigned by the AEG system really are. The general method to prove validity is to compare the scores generated by AEG system with the average of human expert assigned scores [23]. Empirical work in this direction has proved that several AEG systems are valid and reliable. The

correlation results are E-Rater (consistently above 97%), IEA (between 85-91%), PEG (87%), Intellimetric (98%), Automark (between 93-96%) and BETSY (over 80%)[1, 17, 24-27].

Due to the plethora of advantages posed by the use of AES technology, it has been increasingly attracting the interest and attention of schools, universities, testing businesses, researchers and educators [28]. In fact, the automated assessment of essays is regarded by many researchers as the Holy Grail of computer assisted assessment [29].

In the next section, the aim of this thesis is mentioned and explained. Furthermore, specific objectives of this thesis are listed.

## 1.9. Aim of the thesis

The aim of this thesis is to propose an AEG system to automatically perform the assessment of student essays and produce numerical, analytic scores according to the NAPLAN rubric. The input essays are expected to be in electronic texts, preferably in Word documents. The final essay score is obtained by adding up all the analytic scores. So, we aim to automatically grade essays and assign numerical analytic scores for each of the criteria such as spelling, vocabulary and sentence structure. The objectives of this thesis are summarised as follows:

1. To develop an AEG system that is capable of scoring narrative essays according to the NAPLAN rubric, while modeling the linear as well as non-linear relationships between the features and the essay grade.

2. To develop a methodology which enables the AEG system to handle improperly constructed responses as well as those which are properly constructed.
3. To develop a methodology for scoring the ‘spelling’ criterion according to the NAPLAN rubric.
4. To develop a methodology for scoring the ‘vocabulary’ criterion according to the NAPLAN rubric.
5. To develop a methodology for scoring the ‘sentence structure’ criterion according to the NAPLAN rubric.

## 1.10. In the next section, the scope of the thesis is discussed. The areas of AEG that lie beyond the scope of this thesis are adumbrated. Significance of the thesis

To the best of our knowledge, at the time of writing this thesis, no previous work has been conducted on developing an AEG model for grading narrative essays specifically according to the NAPLAN rubric or on developing methodologies for grading the criteria of spelling, vocabulary and sentence structure, specifically according to the NAPLAN rubric. Further, the significance of this thesis can be highlighted as follows:

1. This thesis proposes and implements an AEG system that can grade student essays of the narrative genre, according to the NAPLAN rubric. The features of the proposed

AEG system are: it is genre-specific but not prompt-specific; it can be trained using a relatively small dataset; and it produces optimum results by using minimum available resources and by avoiding resource-hungry processes.

2. This thesis proposes and implements a methodology that can capture both the linear as well as the non-linear relationships between the feature vector and the essay grade. To the best of our knowledge, at the time of writing this thesis, no approach for modeling both the linear as well as non-linear relationships between the feature vector and the essay grade has been proposed.
3. This thesis proposes and implements a methodology to filter out essays which are improperly constructed. Such essays have undesired anomalies or excessive errors in spelling and grammar. To the best of our knowledge, there is currently no existing published literature which explains the handling of improperly constructed essays. In fact, most available AEG systems assume all responses to be properly constructed.
4. This thesis proposes and implements a methodology to grade the 'spelling' criteria according to the NAPLAN rubric. To do so, this thesis proposes algorithms for word classification and automated spelling mark assignment to the essays. To the best of our knowledge, at the time of writing this thesis, no such attempt has been made in literature so far.
5. This thesis proposes and implements a methodology to grade the 'vocabulary' criteria according to the NAPLAN rubric. To do so, this thesis proposes two innovative methods for scoring poor essays and good essays. A rule-based algorithm is proposed for scoring poor essays and a neural network based approach is proposed for grading good essays.
6. This thesis proposes and implements a methodology to grade the 'sentence structure' criteria, according to the NAPLAN rubric. To do so, this thesis proposes a heuristics

and rule-based algorithm for grading poor essays and a neural network-based approach for grading good essays.

In the next section, a pictorial plan of the thesis is provided to assist readers in order to locate a particular chapter or content of the thesis.

## 1.11. Plan of the thesis

Figure 1.2 depicts the plan of this thesis. The purpose of this plan is to provide a road-map for the reader.

Chapter 1 – This chapter provides the introduction to the topic of automated essay grading and defines terms such as essay and automated essay grading. The advantages of AEG systems over manual grading are also outlined. The aim, scope and significance of this thesis are explained.

Chapter 2 – This chapter provides an overview of the existing methods for grading short answer type responses and essays. The working of several semi-automated and fully automated essay grading systems is explained and a critical evaluation of the existing methods is given.

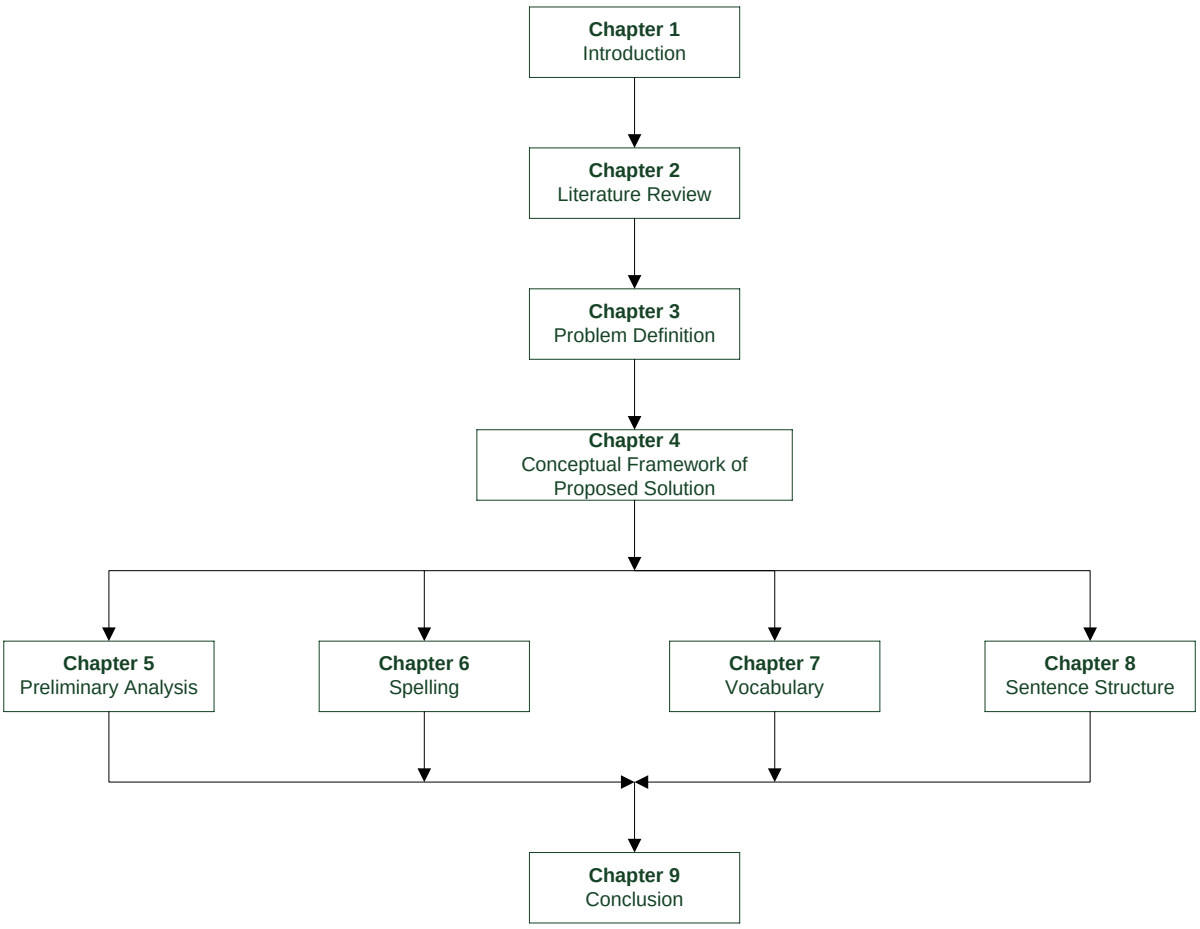



**Figure 1.2: Plan of the thesis**

Chapter 3 –This chapter explains in detail the problem definition that this thesis aims to address, as well as the research methodology that is adopted to solve the identified problem.

Chapter 4 – This chapter explains the conceptual framework of the proposed solution with an overview of the working of each module in the proposed AEG system.

Chapter 5 – This chapter explains the filter process for reducing noise in the essay dataset. Preliminary analysis is carried out to determine if using neural networks for scoring essays is feasible or not.

Chapter 6 –This chapter explains in detail the working of the spelling module. Two novel algorithms are proposed, based on heuristics and rules for word classification and marking

spelling in essays, according to the NAPLAN rubric.

Chapter 7 – This chapter explains in detail the working of the vocabulary module. Two novel algorithms are proposed: one algorithm based on heuristics and rules for grading vocabulary in poor essays and the other algorithm based on neural networks for grading vocabulary in good essays, according to the NAPLAN rubric.

Chapter 8 –This chapter explains in detail the working of the sentence structure module. Two novel algorithms are proposed: one algorithm based on heuristics and rules for grading sentence structure in poor essays and the other algorithm based on neural networks for grading sentence structure in good essays, according to the NAPLAN rubric.

Chapter 9 – This chapter highlights the significance of this thesis and suggests future work related to this thesis. It reiterates the content of the thesis as a whole and summarises this thesis.

In the next section, a summary of the main points discussed in this chapter are mentioned to conclude the chapter.

## 1.12. Conclusion

The purpose of this chapter was to familiarise the reader with the various terms used in the field of AEG, the background and history of this field and to provide a road-map for the rest of the thesis. Accordingly, the importance of AEG systems was explained along with their advantages over the conventional method of the human grading of essays. The criticisms

against the use of such systems were also presented; however, the advantages related to the use of these systems far outweigh the concerns. Hence, these systems have gained widespread use and recognition in multitude and magnitude. The aim and scope of this thesis was detailed, its significance was highlighted and a plan of the thesis was provided.

In the next chapter, a thorough literature review of the existing systems for scoring short answer type responses and essays is presented.

## 1.13. References


[1] S. Dikli, "An overview of automated scoring of essays," *The Journal of Technology, Learning, and Assessment*, vol. 5(1), 2006.

[2] M. D. Shermis and J. Burstein, *Automated Essay Scoring: A cross disciplinary perspective*. Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[3] E. Page, "The imminence of grading essays by computer," *Phi Delta Kappan*, vol. 47, pp. 238-243, 1966.

[4] T. Kakkonen and E. Sutinen, "Evaluation Criteria for Automatic Essay Assessment Systems - There is much more to it than just the correlation," in *Proceedings of the16th International Conference on Computers in Education*, Taipei, Taiwan, 2008, pp. 111-115.

[5] Y.-W. Lee, C. Gentile, and R. Kantor, "Toward Automated Multi-trait Scoring of Essays: Investigating Links among Holistic, Analytic, and Text Feature Scores," *Applied Linguistics*, vol. 31, pp. 391-417, 2010.

[6] (26/07/2011). Available: http://www.ets.org/gre/revised_general/prepare/analytical_writing/score_level_descriptions

[7] E. B. Page, J. P. Poggio, and T. Z. Keith, "Computer Analysis of Student Essays: Finding Trait Differences in Student Profile," presented at the Annual Meeting of the American Educational Research Association Chicago, IL, 1997.

[8] Department of Education, *Narrative Marking Guide 2010, National Assessment Program-Literacy and Numeracy*: Government of Western Australia, 2010.

[9] L. M. Rudner, V. Garcia, and C. Welch, "An Evaluation of the IntelliMetric[SM] Essay Scoring System," *Journal of Technology, Learning, and Assessment*, vol. 4(4), 2006.

[10] D. McCurry, "Can machine scoring deal with broad and open writing tests as well as human readers?," *Assessing Writing*, vol. 15, pp. 118-129, 2010.

[11] E. B. Page, "The imminence of grading essays by computer," *Phi Delta Kappa*, vol. 47, pp. 238-243, 1966.

[12] B. Davies and T. Gralton, "Trial of Automated Essay Scoring:new directions for national assessment in Australia," presented at the International Association for Educational Assessment, Brisbane,Australia, 2009.

[13] C. Yen-Yu, L. Chien-Liang, L. Chia-Hoang, and C. Tao-Hsing, "An Unsupervised Automated Essay Scoring System," *Intelligent Systems, IEEE*, vol. 25, pp. 61-67, 2010.

[14] I. I. Bejar, R. R. Lawless, M. E. Morley, M. E. Wagner, R. E. Bennett, and J. Revuelta, "A Feasibility Study of On-the-Fly Item Generation in Adaptive Testing," *Journal of Technology, Learning, and Assessment*, vol. 2(3), 2003.

[15] E. Cotos and N. Pendar, "Automated diagnostic writing tests: Why? How?," in *Towards Adaptive CALL: Natural Language Processing for Diagnostic Language Assessment*, Ames, Iowa, 2008.

[16] S. Darus, S. Stapa, and S. Hussin, "Experimenting a Computer-Based Essay Marking System at Universiti Kebangsaan Malaysia," *Jurnal Teknologi*, vol. 39(E), pp. 1-18, 2003.

[17] S. Valenti, F. Neri, and A. Cucchiarelli, "An overview of current research on automated essay grading," *Journal of Information Technology Education*, vol. 2, pp. 319-330, 2003.

[18] C.-F. E. Chen and W.-Y. E. Cheng, "Beyond the Design of Automated Writing Evaluation: Pedagogical Practices and Perceived Learning Effectiveness in EFL Writing Classes," *Language Learning & Technology*, vol. 12(2), pp. 94-112, 2008.

[19] J. M. Stalnaker, "The problem of the English examination," *Education Record*, vol. 17, p. 41, 1936.

[20] L. Rudner and P. Gagne., "An overview of three approaches to scoring written essays by computer," *Practical Assessment, Research & Evaluation*, vol. 7(26), 2001.

[21] J. Cheville, "Automated scoring technologies and the rising influence of error," *The English Journal*, vol. 93(4), pp. 47-52, 2004.

[22] L. S. Larkey, "Automatic essay grading using text categorization techniques," in *Proceedings of the 21st ACM/SIGIR (SIGIR-98)*, Melbourne, Australia, 1998, pp. 90-96.

[23] R. E. Bennett and I. I. Bejar, "Validity and Automad Scoring: It's Not Only the Scoring," *Educational Measurement: Issues and Practice*, vol. 17, pp. 9-17, 1998.

[24] J. Burstein, "The e-rater scoring engine: Automated Essay Scoring with natural language processing," in *Automated Essay Scoring: A cross disciplinary approach*, M.

D. Shermis and J. C. Burstein, Eds., ed Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[25] S. Elliot, "IntelliMetric: from here to validity," in *Automated essay scoring: a cross disciplinary approach*, M. D. Shermis and J. C. Burstein, Eds., ed Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[26] T. K. Landauer, D. Laham, and P. W. Foltz, "The Intelligent Essay Assessor," *IEEE Intelligent systems*, vol. 15(5), pp. 27-31, 2000.

[27] T. Mitchell, T. Russel, P. Broomhead, and N. Aldridge, "Towards robust computerized marking of free-text responses," presented at the Sixth International Computer Assisted Assessment Conference, Loughborough University, Loughborough,UK, 2002.

[28] J. Burstein, K. Kukich, S. Wolff, C. Lu, M. Chodorow, L. Braden-Harder, and M. D. Harris, "Automated scoring using a hybrid feature identification technique," presented at the Annual Meeting of the Association of Computational Linguistics, Morristown, NJ, 1998.

[29] D. Perez-Marin, I. Pascual-Nieto, and P. Rodriguez, "Computer-assisted assessment of free-text answers," *The Knowledge Engineering Review*, vol. 24(4), pp. 353-374.

# Chapter 2: Literature Review

## 2.1. Introduction

In this chapter, firstly the key concepts that are used in the field of AEG systems are defined. Then an overview of the existing approaches on the automated scoring of short answer-type responses is given after which several semi-automated AEG systems are discussed in order to provide an overview of methodology of such systems. The semi-automated system 'ePen', currently used by Western Australian Department of Education and Testing (WA-DT) in the marking of the NAPLAN writing assessment is described, after which a comprehensive literature review of fully automated AEG systems is given. These are the systems that utilise the highest level of abstract tools and techniques in the field of text processing and text classification. Each essay grading method is discussed in detail along with its working methodology, its application and its corresponding performance success and shortcomings.

In the next section, the key concepts related to the field of text processing and AEG systems are defined and explained.

## 2.2. Key Concepts

Natural Language Processing (NLP) is the application of computational methods to analyse characteristics of electronic files of text or speech [16]. Natural Language Processing (NLP) is a complex and challenging branch of Artificial Intelligence (AI) [1]. NLP consists of several text analysis processes that can be applied in various research domains such as natural language synthesis, speech recognition, machine translation and information extraction. There are mainly two types of NLP techniques – shallow and complex.

Shallow NLP techniques are those which provide only a surface analysis of text. Illustrative of this type are tokenizer, sentence splitter, stop word removal, stemming, part-of-speech (POS) tagger and chunker. Systems that use these techniques can be ported easily across languages and domains [2]. A *tokenizer* splits a sentence into words called tokens and omits the punctuation marks. A *sentence splitter* splits a block of text into sentences according to the sentence end markers (full stop, exclamation mark and question mark) and displays one sentence per line. *Stop word removal* is a process of removing the most commonly occurring words (such as ‘a’, ‘with’ and ‘the’) which do not contribute much to the content of the text. *Stemming* is a process of roughly chopping off the ends of words to derive the base words. *Lemmatization* is the process of reducing a word to its base word (called the lemma) by removing inflections properly with the use of a vocabulary and morphological analysis of words, rather than by rough chopping [3]. Hence lemmatization produces more accurate results than stemming. A *POS tagger* reads a natural language sentence and outputs the sentence along with tags for parts of speech of each word in the sentence. A *chunker* (also called a ‘shallow parser’) reads a natural language sentence and partitions it into sequences of semantically related words such as noun phrases and verb phrases.

On the other hand, complex NLP techniques allow in-depth syntactic, semantic, discourse and topical analysis of text. Illustrative of this type are syntactic parsers, semantic parsers and rhetorical parsers [4]. In the process of natural language understanding, syntactic analysis and semantic analysis are involved. On the other hand, during semantic analysis, the meaning of the sentence is pictorially represented by using logical expressions [1]. Typically, syntactic analysis involves a process known as 'parsing' and this process is carried out by a parser. According to [5], "a parser identifies syntactic structures, such as subjunctive auxiliary verbs and a variety of clausal structures, such as complement, infinitive, and subordinate clauses" (p.1).

Syntactic analysis involves breaking down text into various constituents and identifying the syntactic dependencies between them. Semantic analysis involves the identification of contextual relationships between the constituents of the text. Rhetorical analysis deals with the identification of discourse elements in the text and the structure of argument development. While these types of analyses are time-consuming and expensive, a thorough understanding of the covert aspects of the text can be achieved by using them. Moreover, systems that use these techniques have limited portability across languages. To improve portability, parsers specific to different languages need to be used in the systems.

Various NLP techniques are used in building automated systems for assessing short answer type responses and essays. In the next section, the methodologies of some automated systems for grading short answer type responses are explained.

## 2.3. Automated Systems for Short answer type responses

As mentioned in section 1.2 in chapter 1, short answer type responses are written in response to questions after a reading comprehension or after a book chapter. These responses consist of about 4-5 sentences where each sentence consists of about 15-20 words. The first step in the general method for automatically scoring such responses is to acquire a model answer key from the tutor. Then NLP techniques such as syntactic and semantic analyses are employed to break down and analyse both the student's response and the model answer key into the smallest possible units. Then, using pattern matching techniques, the two are compared. Depending on the level of similarity, the final score is evaluated and assigned. In this section, we elaborate on some existing automated systems for grading short answer type responses. Grading short answer type responses is a less challenging task than grading free-text essays as shown below.

### 2.3.1. Educational Testing Service 1 (ETS1)

Educational Testing Service 1 (ETS1) was developed by Jill Burstein and Randy Kaplan of the Educational Testing Service (ETS) [6]. This method was developed primarily for scoring short answer type responses, consisting of sentences of 15-20 words.

To develop this system, a concept-based lexicon and a grammar rules database was manually built from training data. Firstly, the training essays were parsed with the Microsoft NLP tool. Then suffix stripping and stop word removal was performed manually. This produced a lexicon. Grammar rules were constructed for each category of answers in order for the system to be able to detect all possible paraphrases of the correct answer. To grade new answers, the following steps are followed: the answers are parsed through the phrasal node extraction par-

ser to obtain different phrases in the text; this output is collapsed into a generic XP phrase type; and the sentence is compared against the grammar and lexicon for matches.

Success: The authors claim an accuracy of 80% when marking a test set and 90% when marking both the training and test set, with the test set containing the training set. In another evaluation that used an improved lexicon, an accuracy of 93% was reported for marking the test set and 96% for marking both sets.

Shortcomings:

1. This system is suitable only for grading short answer-type responses. Hence, it cannot be used for grading essays.

2. Since this is one of the earliest AEG systems, it requires a lot of manual calibration work and pre-processing.

### 2.3.2. Conceptual-Rater (C-Rater)

C-rater is the acronym for Conceptual rater. It was developed by ETS for scoring short answer responses, such as those at the end of a textbook chapter [7].

For every question, C-rater uses a single correct answer called the answer key. To score the student response as correct or incorrect, the concepts presented in the student response are compared to the concepts in the answer key. Using NLP techniques that are used in an essay grading system E-rater (described in section 2.6.2 in this chapter), an analysis of the predicate argument structure is carried out by detecting the logical relations between the various syntactic components. It is claimed that C-rater is robust enough to deal with syntactic variations in the sentence structure, words in different inflections, spelling mistakes, synonyms and all possible paraphrases that can be developed from the answer key [8].

Success: In an evaluation of C-rater in a university virtual learning program, it achieved over 80% agreement with the score assigned by an instructor. Furthermore, in a large-scale assessment, when C-rater was used to score 170,000 short answer responses to 19 reading comprehension prompts and 5 algebra prompts, an accuracy of 85% was reported [2].

Shortcoming:

The major shortcoming of C-rater, like ETS (1), is that it is suitable only for grading short answer-type responses. Hence, it cannot be used for grading essays.

### 2.3.3. Automated Text Marker (ATM)

The major shortcoming of IEA as mentioned in section 2.3.1 above is that it is a 'bag of words' approach. In order to overcome this, Automated Text Marker (ATM) was developed in 2001 at Portsmouth University, U.K [9]. ATM considers word order in the student response and can be used to assess short answer type responses [10].

In this method, Information Extraction (IE), which is a new type of NLP technique, is used and it intelligently skims the input text searching for specific concepts rather than doing an in-depth analysis [11]. To analyse the content, using IE and semantic analysis, an examiner's model answer is broken down into smallest possible units by extracting various concepts and their underlying dependencies and relations. Similarly, the student's response is broken down into the smallest possible units and then using pattern matching techniques, the two are compared. Depending on the similarity between them, a summative score is assigned for content. To analyse the style, grammar checking of the student's response is performed by using syntactic analysis. The final score for the student's response is the sum of the scores assigned for content and style.

The success of this system is not known because results obtained from this system are not yet published [2].

Shortcomings:

This method is suitable only for short answer type responses to fact-based questions such as in Biology and Psychology. Hence, it cannot be used for essay grading.

### 2.3.4. Automark

[12] propose Automark which uses the IE technique similar to ATM. Automark is aimed at the robust computerized marking of free-text answers to open-ended questions. It was initially an academic work but later on, in 2002, it became commercially available by the company Intelligent Assessment Technologies. It can be used to grade short answer type responses.

A number of processing modules are employed in the software system Automark. These modules are aimed at achieving robust marking despite errors that may appear in spelling, syntax and semantics. The marking process has a number of stages. In the first stage, the incoming text is pre-processed to standardize the input in terms of punctuation and spelling. Then, the standardized input is processed through a sentence analyser that identifies the main syntactic constituents of the text and analyses how they are related. These syntactic constituents are used by the pattern matching module to find matches with the mark-scheme templates where each template specifies a correct or incorrect answer for a particular question. A single mark is assigned to each correct template and finally all the marks are added up. The development of templates is an offline process and is achieved using a system configuration interface that is custom written. The representation of templates is done in such a way that they are robust enough to handle multiple variations in the input text.

Finally, the feedback module processes the result of the pattern matching module and assigns the feedback which is in the form of a mark. However, the authors claim that more specific feedback is possible.

Success: Automark has been tested on eleven-year-old pupils in the National Curriculum of Science. The response required from the students was of four types, in increasing order of linguistic complexity: single word generation, single value generation, generation of a short explanatory sentence, and description of a pattern in data. 120 responses were randomly selected for each type of question. 40 pupils were the same at each level for each of the four types. In this experiment, the authors achieved a correlation between 93% and 96% with human graders.

Shortcomings:

1. According to [2], it cannot identify spelling mistakes correctly, it cannot analyse the sentence structure, it cannot identify an incorrect answer and cannot assess a response that has content other than the content provided in the mark scheme templates.

2. Since it was developed for grading short answer type responses, it cannot be used for the automated evaluation of essays.

In the next section, the classification of essay grading systems is mentioned. Examples of essay grading systems of each type are provided and their methodologies are explained along with their success and shortcomings.

## 2.4. Types of Essay Grading Systems

Existing essay grading systems can be broadly classified into two types: semi-automated and fully automated, depending on the amount of human intervention required during the grading process. A semi-automated essay grading system usually provides only a visual user interface to display the essay and some annotation buttons for selecting the score and feedback. The actual grading task is to be performed by a human grader because the semi-automated system is not equipped with the algorithms required to perform grading automatically. Some of the existing semi-automated essays grading systems are detailed in the next section.

## 2.5. Semi-Automated Essay Grading Systems

In the existing literature, some of the semi-automated essay grading systems are Methodical Assessment of Reports by Computer (MARC), Markin32, Student Essay Viewer and ePen. The methodology and working of these systems is explained in this section.

### 2.5.1. Methodical Assessment of Reports by Computer (MARC)

MARC is a computer-based semi-automated grading tool and was developed for ensuring uniformity in the correction of student written reports [13]. The main motivation behind developing the tool was to ensure that five tutors teaching the same course in a university corrected student reports using the same criteria. The methodology adopted to achieve this objec-

tive is as follows: after the tutor goes through the student report, he fills in a three-page assessment sheet generated by the computer. This sheet contains the same assessment questions for all the students. Hence, the correction of reports results in a uniform criteria evaluation.

### 2.5.2. Markin32

Markin32 is a computer-based tool for annotating essays and providing feedback on them [14]. This tool was developed to reduce the correction workload of tutors. Students write their essays using a Word processor and submit them via email. The tutor opens their essays using the Markin32 tool and corrects them while annotating them on the computer. The tool displays the essay in a window along with custom-designed annotation buttons, one for each type of error. For example, the button 'Sp' denotes 'spelling error', and so on. Furthermore, by using this tool, the tutor can provide feedback either on his own or by choosing from the database provided. Then the annotated essay and the feedback can be sent to the student electronically.

### 2.5.3. Student Essay Viewer

[15] developed a visualisation tool that automatically highlights the argumentation cue phrases in a student's essay. The main idea of this tool is to help teachers visualise arguments in the essay so that they can then give the score and feedback depending on what was covered and not covered in the essay content. The developers of this tool compiled a database of the different types of argumentative phrases and patterns and assigned each of them into one of nine pre-defined categories such as: *reporting* (phrases containing words such as 'X discusses', 'Y warns'), *positioning* (phrases containing words such as 'I accept'), *connectors* (links between propositions, for example, 'therefore', 'in fact', and 'however') and so on.

In the user interface of this tool, the essay is displayed in a window and all the argumentative phrases are highlighted and their respective annotations are displayed as semantic tags. The

authors assert that this tool can be used by teachers to quickly check the type of content and extent to which the essay covers the required argumentation. Additionally, students can use this tool for improving their writing. They can revise their drafts in accordance with the feedback regarding incomplete content.

### 2.5.4. ePen

This computer-based tool, developed by Pearson Technologies, is currently being used by WADET. It is a visualization tool that displays a student essay in a window on a digital screen, along with the NAPLAN score range for each evaluation criteria, beside the essay. For example: the criterion 'spelling' has a score range of 0-6. To correct the essay, the marker can read the essay on the computer and assign scores by clicking on the relevant criteria range. Further, the marker can provide any comments about the essay in own words.

The next section outlines the various types of fully automated essay grading systems and their methodology. The performance success and the shortcomings associated with each system are also highlighted.

## 2.6. Automated Essay Grading Systems

Automated Essay Scoring (AES) is defined as computer technology that evaluates and scores written prose [1,5]. An automated essay grading (AEG) system is a computer-based system that is designed and developed in such a way that the computer can automatically assess essays and assign appropriate scores to them, with minimum or no human intervention required

during the grading process. The techniques used in the systems can be natural language processing or statistical or a combination of both.

As mentioned earlier, Natural Language Processing (NLP) is the application of computational methods to analyse characteristics of electronic files of text or speech [7]. All the methods that employ language processing techniques, including parsing, pre-processing and lexical-similarity comparisons, can be classified under this category [16, 17]. However, it is not really possible to assign the grade only by analysis of the essay text. After analysis, the values obtained during analysis are to be evaluated using some kind of statistical technique in order to assign a final score to the essay. Hence, most of the available systems use a combination of both NLP and statistical techniques.

Statistics is the science of making effective use of numerical data relating to groups of individuals or experiments [32]. Several statistical learning methods such as regression models including linear regression and step-wise linear regression, Latent Semantic Analysis (LSA), Text Categorisation Techniques such as nearest neighbour classifiers, Bayes belief networks; and neural networks have been applied to solve the problem of automated text categorization in the last few years [18].

'Hybrid methods' is defined in this thesis as methods that are based on a combination of NLP and statistical techniques. In the existing literature, almost all methods can be classified as hybrid methods. However, a number of AEG systems employ the Latent Semantic Analysis (LSA) technique as the backbone of the system. LSA is defined as "a statistical model of word usage that permits comparisons of the semantic similarity between pieces of textual information" ([19], p.2). All methods that are based on the LSA technique can be classified into LSA-based methods. Furthermore, a number of AEG systems employ Text Categorisation Techniques (TCT) techniques at the backbone of the system. Text Categorisation Techniques

are used ‘to train binary classifiers to distinguish ‘good’ from ‘bad’ essays, and use the scores output by the classifiers to rank essays and assign grades to them’ ([20], p.90). All methods that are based on text categorization techniques can be classified into TCT-based methods. Further, there are a number of AEG systems that employ miscellaneous techniques and hence can be classified into miscellaneous technique-based methods. Therefore, for the purpose of explaining the methodology of each system in detail, the existing AEG methods have been broadly categorized into four categories: a) hybrid methods; b) LSA-based methods; c) TCT-based methods; and d) miscellaneous technique-based methods.

In the existing methods, the AEG systems PEG, E-rater, E-rater V.2, Criterion; Schema Extract Analyse and Report (SEAR), Intelligent Essay Marking System (IEMS), Paperless School free text Marking Engine (PS-ME), Intellimetric, My! Access and an AES system for CET4 can be classified as hybrid methods. The methods Intelligent Essay Assessor (IEA), Automatic Essay Assessor (AEA), Jess, MarkIT and an AEG system using Generalised LSA can be classified as LSA-based methods. Further, the methods TCT, Bayesian Essay Testing Scoring sYstem (BETSY), CarmelTC and two AEG systems using k-Nearest Neighbour (kNN) can be classified as TCT-based methods. Finally, an AEG method using connections between paragraphs, an AEG method using literary sememes, an AEG system using unsupervised learning and an AEG system using a modified BLEU algorithm can be classified as miscellaneous technique-based methods. Each of these methods is explained in detail below.

### 2.6.1. Hybrid methods

AEG systems which are based on a combination of NLP techniques and statistical techniques can be classified as Hybrid methods. Quite obviously, these systems benefit from the advantages of both types of techniques. Hence, the performance of these systems is generally

better than the AEG systems in the other categories. In this section, the methodology of hybrid AEG systems is explained.

#### 2.6.1.1. Project Essay Grade (PEG)

Ellis Batten Page of Duke University in USA developed the Project Essay Grade (PEG) in 1966 [21]. His aim was to reduce the teacher's workload and to improve the assessment process [22]. He coined two new terms: 'trins' meaning the intrinsic variables within the essay and 'proxes' which refers to the approximation (made by the computer) of the intrinsic variables. The underlying assumption in PEG is that the quality of the essay is displayed by the proxes, which are an indirect measure.

Firstly, a set of manually pre-graded essays is used to predict the proxes that best influence the essay grades. After determining the proxes, they are transformed and used in a standard multiple regression along with the grades to calculate the regression coefficients. Some of the proxes are: average word length, essay length in words, number of commas and prepositions, sentence length and fourth root of number of words. To grade a test essay, the values of its proxes are calculated. Then by using these and the regression coefficients determined earlier, the grade of the test essay is assigned.

Originally, PEG was a purely statistical method but in the revised version released in the 1990s, grammar checkers and part-of-speech taggers have been incorporated [23]. Hence, it is more suitable to classify it as a 'hybrid' system. Recently a web-based interface has also been developed for PEG [24]. It currently includes assessment of content, style, mechanics, organization and creativity in essays.

Success: PEG achieved a regression correlation of up to 0.87.

Shortcomings:

1. The system can be easily tricked and fooled by students seeking to obtain a better grade, for example, by writing a very lengthy essay [1].

2. For every new essay question, PEG needs to be run to build the regression equation specific to that question. Hence, it is prompt-specific but not a generic model.

3. There is not enough evidence that this system can be used for grading narrative essays.

4. This system assumes that the relationship between essay features and its grade is linear, which is not necessarily true.

5. This system assumes that the essays to be graded are properly constructed responses.

#### 2.6.1.2. E-rater

E-rater, originally called Computer Analysis of Essay Content, was developed by Burstein et al. at the Educational Testing Service (ETS), USA, in 1998 [7]. ETS conducts large scale assessments like GRE (Graduate Record Examination), TOEFL (Test of English as Foreign Language) and GMAT AWA (Graduate Management Admissions Test Analytical Writing Assessment) [5].

E-rater is a hybrid feature technology that takes into consideration both the content and style of the essay. It employs several NLP techniques and incorporates five modules in the essay grading system [7]. The three main modules are: the syntactic module used for syntactic structure analysis; the discourse module used for argument analysis; and the topical analysis module for topical content analysis. In the syntactic module, E-rater tags each word for part-of-speech using the Brill Tagger, then uses a syntactic “chunker” to extract phrases and finally, assembles the phrases into trees based on sub-categorization information for verbs. The essay is parsed using the Microsoft Natural Language Processing tool (MSNLP) to assess the syntactic structure of the essay. For each sentence in the essay, the number of complement

clauses, subordinate clauses, infinitive clauses and relative clauses are calculated as a measure of syntactic variety.

In the discourse module, E-rater looks for 'rhetorical' surface cue words and structures. After identifying the different terms that mark the beginning of the argument, the continuation and the termination of the argument, the E-rater's program Argument Partitioning and Annotation (APA) reproduces the essay by partitioning it "by argument" instead of "by paragraph" as was originally written by the author [7]. The output of the APA is used by the next module to evaluate the content of individual arguments [25, 26].

In the topical analysis module, E-rater uses two methods: EssayContent (based on word frequency) and ArgContent (based on word weight) [26]. For both the methods, essays in the training set are converted into vectors of word frequencies which are then transformed into word weights. In the method EssayContent, to assign a score to the test essay, it is first converted into a weight vector and then the cosine correlation between the weight vector of the test essay and the vectors of the training essays is determined. The score for the test essay is obtained by calculating the weighted mean of the scores of the six training essays who have the lowest cosine correlation with the test essay. In the method ArgContent, six 'supervectors' are constructed, one for each score from 1-6, using the training essays in each score category. The test essay is assessed one argument at a time. A score is assigned to every argument of the test essay by calculating the lowest cosine correlation between the argument weight vector and the six 'supervectors'. The final score assigned to the test essay is the mean of the scores of all the arguments [27].

Model Building: This module is a program that executes a step-wise regression. The outputs from the syntactic module, discourse structure module and the topical analysis module are fed as inputs to this program. It automatically extracts the most predictive features of the training

essays and outputs this along with the regression weights. It uses about 60 feature scores and then using step-wise regression, only 8-12 features are selected for any given prompt [23].

Scoring: The regression equation is used to compute the final essay score. The product of each of the regression weight and its associated feature value is calculated and these are then summed to obtain the final score for the test essay.

Success: Initial training and evaluation had a level of correlation with human graders between 87% and 94% across 15 test questions [26]. After undergoing further improvements, E-rater was used to score essays for GMAT AWA in 1999 and since then, it has been used to score 750000 high-stakes essays where the agreement between human graders and the computer was found consistently above 97% [7]. It is also reported that E-rater scores show a 3% discrepancy with a single human grader which is the same between two single human readers. Further, it is reported that E-rater is able to adapt very well to different topical domains and populations of test-takers [26].

When an essay is too short or too different from the other essays, the system flags the essay and issues an advisory message [2].

Shortcomings:

1. It is very complex and requires a lot of training.
2. It can detect only two types of improperly constructed essays – essays which are too short or too different from other essays. It does not consider other types of improperly constructed responses.
3. It requires a considerable number of essays to be scored manually by human graders. For example, the Web-based version of E-rater, called the Criterion, requires 465 human-scored essays to build the scoring model.

4. For every new essay question, the E-rater system needs to be run to build the regression equation specific to that question. Hence, it is prompt-specific but not a generic model.
5. Jill Burstein herself critiqued that E-rater regards an essay as a 'bag of words'. It means that the system can be tricked by writing a lot of words which might be correct but do not contribute to the line of argument in the essay [28].
6. Validity tests on E-rater reported that it sometimes rewards an essay with a higher score than a human rater would. Hence, it was proposed that the E-rater system be always used in conjunction with a human rater [29]. If the score assigned by E-rater and the human rater has a discrepancy of more than one point, then a third rater is called in to assess the essay.
7. Although E-rater has a comprehensive essay analysis model, it can be used only for holistic scoring.

#### 2.6.1.3. E-rater V.2

An improved version of E-rater, called the E-rater version 2, was released in 2003 [23]. This version uses only 12 features which are claimed to be more precise representatives of the linguistic quality of the essay. Values of these features are extracted for every prompt and empirically derived weights are assigned to them, depending on the prompt. Some of the features are the ratio of grammar errors to the total number of words, the ratio of usage errors to the total number of words, the average word length, number of discourse units detected in the essay, the total number of words and the ratio of different content words to the total number of words. The scoring model is developed in a similar way to E-rater and is then used to score a test essay.

Shortcomings:

1. Like its predecessor, E-rater V.2 needs to be trained specifically for every new prompt. This means that it is a prompt-specific AEG model, whereas, a ‘generic’ model that can be applied to any prompt within a specific genre is more desirable [23].
2. This AEG system determines various word counts but does not consider the content of the words. So, the system can be fooled by writing a lot of complex words that fail to add meaning to the passage [30].
3. It does not model the non-linear nature of the relationship between essay features and its grade.
4. It can be used only for holistic scoring but not for analytic scoring.

#### 2.6.1.4. Criterion (Web-based application of E-rater)

ETS Technologies, Inc., a subsidiary of ETS, developed a web-based fully automated essay scoring system called Criterion. Criterion relies on *E-rater* as its back-end and *Critique* for writing analysis tools [1]. In other words, Criterion is an ‘E-rater on the web’ and is used both as an instructional tool for instructors and students as well as for writing assessment in institutions. Instructors and students can submit an essay to the online writing evaluation service and within seconds, they will receive the score of the essay.

The initial version of Criterion was able to provide only the score for an essay but later on, a feedback component, called the ‘advisory component’, was appended to the model. This module works independently from the E-rater score evaluation module. Further, it makes use of statistical measures and provides information related to brevity, repetitiveness of response and off-topicness of essay response [7]. The feedback provided is:

a. The number of words in the essay is counted and a comment is given if the essay is too short.

b. If the essay is overly-repetitive, a comment is given that the student should use more synonyms.

c. If the content of the essay is not similar to other essays written on the topic, a comment is given that probably the essay is off topic.

Success: Currently TOEFL and GMAT use this writing evaluation service for low stakes writing evaluation of practice essay tests. Additional facilities for students are the electronic portfolio (for storing drafts) and writer's handbook (containing definitions and examples of language usage) [1]. The central control of the program lies with the teacher such that the teacher can choose what features to enable or disable for the students. Moreover, they can include their own feedback apart from using the feedback from Criterion.

Shortcomings: The following are the limitations of Criterion:

1. Since it uses E-rater, it provides only holistic scores and cannot be used for analytic scoring.

2. A study undertaken with 46 students at the Universiti Kebangsaan, Malaysia, proved that feedback provided by Criterion is not quite informative and is useful only to an extent [31].

3. It is reported that Criterion cannot detect illogical or inventive essays. Further, it is vehemently argued that Criterion should not be used in academic institutions in particular and for instructional use in general [28]. The author's argument in [28] is based on the premise that a writer writes to convey meaning to readers and the meaning cannot be represented by a formula based on a static set of features.

4. It considers only three types of improperly constructed responses.

#### 2.6.1.5. Schema, Extract, Analyse and Report (SEAR)

Schema, Extract, Analyse and Report (SEAR) is an automated essay grading system developed by James R Christie at The Robert Gordon University, Aberdeen [32]. It was developed as the author's PhD research work. In SEAR, the marking style process is very similar to that of PEG and is based on the 'common' metrics found in essays. This process comprise of the four following steps:

1. The 'common' metrics or features associated with the essays are pre-determined [32]. However, since there is no standard set of metrics, the best thing is to devise a suite of standard sets, one set of metrics for a particular essay type.

2. From the essay set, a sample of essays should be marked manually. Statistically, the minimum sample of essays should be twice the number of metrics that are determined in the first step.

3. A weighted linear function is obtained by processing the subset of essays and adjusting the weight of each metric until an agreement is obtained between the human and computer marking.

4. Finally, all the remaining essays are processed.

To measure the content of the essays, the two terms "usage" and "coverage" are used to study the relationship between each essay and the schema. In both these terms, high is good and low is bad. 'Usage' is a measure of how much of each essay is used while 'coverage' is a measure of how much of the schema is 'used' by the essay under examination. The schema can be stored as a simple data structure as in a COBOL program, and is prepared once and can be easily revised. Further, it does not need to be prepared in advance and needs neither

‘training’ nor ‘calibration’. An essay that is high in both ‘usage’ and ‘coverage’ implies that it is very well written.

Shortcomings:

1. It can assess the content of technical essays only so it cannot be used for grading essays from the narrative genre.
2. It assumes that the relationship between the essay features and its grade is linear, which is not necessarily true.
3. It makes an assumption that the essays to be scored are properly constructed responses, which is not necessarily true.
4. This system assigns holistic scores. Hence, it cannot be used for analytic scoring to assign scores individually for criteria such as spelling, vocabulary and so on.

#### 2.6.1.6. Intelligent Essay Marking System (IEMS)

Intelligent Essay Marking System (IEMS), developed at the NGEE ANN Polytechnic in Singapore, is based on a Pattern Indexing Neural Network technique called the Indextron [33]. Indextron is defined as a specific clusterization algorithm, which is not a neural network, but can be implemented as a neural network. The main idea behind IEMS is that answers that contain similar word patterns obtain similar scores and hence, can be grouped to form clusters where each cluster consists of answers of same score [2]. The Indextron aims to overcome slow, non-incremental training that is typical of an artificial neural network. The Indextron is similar to RAM where new data can be added to it without training as long as the memory size is large enough. Further, the system can be run on an ordinary PC.

IEMS can be used both as an assessment tool and for diagnostic and tutoring purposes in many content-based subjects like psychology, history and other non-mathematical subjects. It evaluates features related to both content and style [34]. This system can be embedded in an

intelligent tutoring system and can therefore help students write better by grading papers fast and providing immediate feedback to the students, indicating the areas where they did well in the essay and the areas where they should do better.

Success: The system was tested using essays by 85 students in third-year mechanical engineering doing a module on Project Report Writing and a correlation of 0.8 was obtained as a result of the test.

Shortcomings:

1. This system assumes that the essays to be graded are all properly constructed responses, which is not true.
2. There is no evidence that this system can be used for grading narrative essays.
3. This system assigns holistic scores. Hence, it cannot be used for analytic scoring to assign scores individually for criteria such as spelling, vocabulary and so on.

#### 2.6.1.7. Paperless School free text Marking Engine (PS-ME)

PS-ME stands for Paperless School free text Marking Engine and is an AEG system based on the objectives of Bloom's Taxonomy [35]. It employs several NLP techniques and functions as a back-end service to a Web-based Learning Management System (LMS). It can be used in the assessment of student essays and short answer type responses [36]. Bloom identified a six element taxonomy, consisting of three main components: knowledge, understanding (comprising the four categories comprehension, application, analysis and synthesis) and evaluation [35].

PS-ME can be set up in two modes: summative assessment and formative assessment [36]. The process of setting it up for a particular task is as follows: select master texts; have a hand-marked sample (can be as few as 30 - this action needs to be done only once in order to derive

the right weightings for the parameter values. Once the weightings are calculated, they can be re-used when that particular task is set again); run the same sample through the marker and perform regression analysis; upload the result to the server.

A master text is compiled from various sources such as textbooks, encyclopaedias and web-sites. When a student essay is submitted for assessment, it is sent to the server along with the information about the task, in order to identify correct master texts for comparison. The task is defined via a number of master texts which are relevant to the question to be answered, and 'negative' master texts which contain a set of false statements and common misconceptions by students. Every comparison with the master texts is given a weighting, negative in the case of 'negative' master text. Weights are derived during the initial training phase. The student essay is compared to the relevant master texts and a number of parameters reflecting knowledge and understanding are derived. For the purpose of evaluation, the individual parameters thus obtained are combined in a numerical expression to calculate the grade of the essay, typically a National Curriculum grade or a GCSE level. Additionally, the system can also provide formative feedback (from a pre-compiled comment bank relevant to the task) to the student regarding his or her performance in different areas of the subject.

Success: It is one of the few AEG systems that take negative answers into account while developing the grading mechanism. However, to the best of our knowledge, evaluation results for PS-ME have not been published yet [2].

Shortcomings:

1. It cannot be used to grade essays in real-time because of its processing requirements.
2. Selection of appropriate master texts seems to be critical for the successful evaluation of essays. However, there are no clear set rules for the selection of master texts. Hence, the selection of master texts will require a new skill and precision.

3. Material from graduate and postgraduate levels has been difficult to automark with accuracy.
4. The most important drawback is that the student is expected to check his or her spelling and grammar because incorrect spellings and wrong grammar can throw the system out.
5. The system cannot handle improperly constructed responses.

#### 2.6.1.8. Intellimetric

Intellimetric was developed by Vantage Learning and released in 1998, after a decade of research and development. It is a hybrid model employing Artificial Intelligence (AI), NLP and statistical technologies [37]. Intellimetric provides an overall holistic score and individual scores in five domains: focus and meaning, organization, content and development, language use and style, mechanics and conventions.

Intellimetric employs AI to emulate the process of scoring carried out by expert human scorers. According to the developers, it 'internalizes the pooled wisdom of many expert scorers'. The system must be initially "trained" with a set of previously scored essays and the "known score" marker papers for each score point are used to develop the rubric and the pooled scores assigned by the human scorers [38].

For essay scoring by Intellimetric, firstly, the essay is parsed and the syntactic and grammatical structure of each sentence in the essay is tagged. Several patented technologies are applied to scrutinize the text. As a result, more than 500 linguistic and grammatical features of the essay are tagged. Then, the data is coded to support the development of many mathematical models to replicate multiple judges. The information from different models is integrated using a proprietary optimization technique and finally, a single score is assigned to the essay. In the third step, a new set of essays are presented to the system for scoring [1].

Intellimetric can be run in two modes: the instructional mode and the standardized assessment mode. It provides an essay score and detailed feedback about various features of the essay. Intellimetric differs from the other essay grading methods, in that, Intellimetric is not manually given a set of features beforehand, instead the method derives the characteristics pertaining to each score point when it is presented with the known score essays.

Success: The developers claim that Intellimetric scores agree with expert human scorers approximately 98-100% of the time [39] and that it is capable of flagging "fabricated" essays that are written to fool the system [38]. It can identify essays that have content too similar to other essays that the system has encountered before or if other linguistic problems are evident, in which case, Intellimetric returns a message saying that the essay cannot be scored along with the reason why. The authors claim that Intellimetric is able to learn from its own mistakes and can improve its accuracy as the system is used increasingly.

Shortcomings: The following are the shortcomings of Intellimetric:

1. No significant studies have been done to "beat the system" and to demonstrate the weaknesses of this system.
2. It is reported that Intellimetric has a tendency of awarding higher scores to essays than human raters [40]. The implication is that if Intellimetric is used in a college placement exam, it would assign a grade of 'pass' to an essay even though the essay may reflect that the student is not yet college ready. This can result in the student being placed in a college and then find it difficult to keep up with the college coursework.
3. It is reported that Intellimetric cannot detect off-topic essays satisfactorily [38].
4. It considers only three types of improperly constructed responses.

#### 2.6.1.9. My! Access™ (Web-based application of Intellimetric)

My! Access™ is an online portfolio-based writing instruction program by Vantage Learning and is based on the Intellimetric system [38]. My! Access™ can provide holistic as well as analytic scores and can be used for genres or essays such as narrative, persuasive, literary and informative. Apart from providing scores and feedback, My! Access™ provides a vast variety of writing assistance features. My! Access™ is able to assign essay topics automatically to students [1]. More information about this system can be accessed at their website [39].

An extensive trial carried out by [41] in Australian schools reported in favour of the use and effectiveness of My! Access™. However, when it was implemented in a thorough trial over a period of six months in Taiwan, it was perceived less favourably by students [42]. Additionally, it seemed to limit their learning of writing because it imposed restrictions on the topical content, organizational structure and discourse style used by them. Furthermore, when the essay was flagged as "off-topic", students were given no explanation as to why it was off topic. In fact, the feedback was criticised as "vague", "abstract", "unspecific", "formulaic" and "repetitive".

Teachers enjoyed the sole power of enabling and disabling the features according to the year level of students and the type of assessment being undertaken. Teachers could choose either pre-built essay prompts provided with My! Access™ or they could design own essay prompt. However, My! Access™ can grade only those essays on which it has been trained.

#### 2.6.1.10. AES system for College English Test band (CET4)

An AES system for grading essays for the College English Test band 4 (National English level test in People's Republic of China) is elucidated in [43]. Using a combination of NLP and statistical techniques, this method can evaluate the essay score related to surface features, grammar, syntactic correctness of sentences and the off-topicness of the essay. Various sur-

face features as used in TCT are determined. For grammar checking, both the bigram word model and the part-of-speech tagged model are considered. For sentence correctness, the part-of-speech sequences of the essay are compared with that of the training set. Then, rules pertaining to the most common errors in CET4 essays are constructed and checked. To detect whether the essay is on topic or not, two methods are used: (1) keywords on the topic are searched for in the essay; and (2) the cosine of the content vector of the essay is compared for similarity with the content vector of the topic. Finally, using linear regression, the final score of essay is computed.

Success: Their system reports a precision of 70.1% with 2-score deviation.

Shortcomings:

1. The underlying assumption that the essay features are linearly related to the essay score is not necessarily true.
2. The authors of this system provide no evidence as to essays of which writing genre are being graded using this method.
3. This system has been designed for grading essays written by English as Second Language (ESL) learners. Hence, it might not be suitable to use this system to grade essays written by native language users for whom English is the first language.
4. It does not propose a methodology to handle improperly constructed responses.

### 2.6.2. Latent Semantic Analysis (LSA)-based methods

LSA was originally proposed by a psychologist named Landaueur and his colleagues [44]. It was developed for the purpose of indexing documents for information retrieval. Owing to the success of LSA in document indexing, it has been employed in the field of AEG with little variation/modification to perform the task of the automated grading of essays. AEG systems that are based on the LSA technique are classified in this category. In this section, the work-

ings of AEG systems such as IEA, AEA, Jess, MarkIT and using G-LSA that are based on LSA-based methods are explained.

#### 2.6.2.1. Intelligent Essay Assessor (IEA)

The Intelligent Essay Assessor (IEA), a commercial AEG system produced and marketed by Pearson Knowledge Analysis Technologies (PKT), is based on the Latent Semantic Analysis (LSA) technique. The original LSA technique was modified to serve the purpose of AEG.

The three main modules in IEA are:

- Content module. This is the most important module, as when using the LSA technique, the content of the essay is scored depending on tk similar calibration essays and the domain relevance is scored depending on the length of the essay vector.
- Mechanics module. The punctuation and spelling in the essay are assessed and scored in this module.
- Style module. The coherence and grammar in the essay are evaluated in this module. Coherence is measured from the LSA value for context-relatedness and the grammar value is measured by a comparison of the essay sentences with the student's essay sentences [45].

LSA is defined as "a statistical model of word usage that permits comparisons of the semantic similarity between pieces of textual information" ([19], p.2). To train the system, a domain text is required which depends on the type of technique. In the holistic technique, a set of pre-graded essays (300 or more) on the same topic are used, whereas in the gold standard technique, the teacher can provide a “model answer” to be used. The domain text is represented by a two-dimensional matrix, with the words representing the rows and the columns representing sentences, paragraphs and other subdivisions of the essay in which the words occur. Each cell in the matrix represents the frequency of the word in each context. This matrix is

then transformed and decomposed into three smaller dimension matrices using the matrix algebra technique called Singular Value Decomposition (SVD) [46, 47]. The decomposition is done in such a way that when the three matrices are multiplied, they should be able to produce the original matrix. As a result of applying the SVD technique, new relationships become manifest, apart from the previous relationships that existed between the words in the sentences [48].

For an essay to be graded, first the reduced dimensional semantic space is developed as explained. Then, to determine the grade of the essay, the lowest cosine correlation is calculated between the semantic space of the essay and the semantic space of the domain text.

Success: IEA is capable of detecting synonyms and paraphrased sentences. It allows students to revise their essays and resend them. Experiments using IEA on GMAT essays resulted in percentages between 85-91% in agreement with the human graders. Further, it is reported that the grading performance of LSA is about as reliable as human graders [19]. The advantage of using cosine correlation to assign the grade to an essay is that it does not depend on the length of the essay. Further, IEA can be applied to essays in any language because LSA is based on machine learning, not on language-dependent rules. It is claimed that IEA is a low cost technique, provides plagiarism detection and is able to give immediate feedback, within 20 seconds, to the student about the writing and the content of the essay. Moreover, IEA is capable of detecting off-topic essays and essays that are too similar to other essays that the system has graded before [9, 49-51]. PKT claims that IEA can be trained only on a set of 100 pre-scored essays, which is a reasonable number when compared to the 300-500 essays required by other systems [1].

The shortcomings of this method are:

1. LSA makes no use of word order and is based on the ‘bag of words’ approach.

2. It is very computationally intensive and the computations are very cumbersome because of the extremely large matrix sizes. For this reason, IEA is not available as stand-alone software as the hardware required for this method is beyond the capabilities of a desktop computer [45,51].

4. LSA makes a subtle assumption that the essays to be scored reflect a properly constructed response, which is fundamentally flawed [45].

5. According to the developers of IEA, it cannot be used to evaluate essays of the narrative genre [1].

6. In an experimental trial of IEA at Curtin University of Technology, Perth, it was reported that IEA is practicable only when very large numbers of essays (typically thousands) are to be graded because the effort involved in formatting and manual grading essays for the semantic space, and the setup costs, are too great when only a few hundred essays are to be graded. Also, since the system was not run at the site of the trial but at a remote site in the USA, there is some lack of control and a potential security risk [52].

#### 2.6.2.2. Automatic Essay Assessor (AEA)

The Automatic Essay Assessor (AEA) was developed by Kakkonen in 2004, at the University of Joensuu, Finland [53,54]. Similar to IEA, AEA is based on the Information Retrieval (IR) technique, Latent Semantic Analysis (LSA). According to [55], content-based grading can be performed in two ways: (1) by comparing an essay to human-graded essays and assigning the grade based on the grades of the *k* nearest neighbor essays; and (2) by basing the grading on both human-graded essays and course content.

AEA uses learning course materials such as course textbooks and lecture notes; and a set of pre-scored student essays for a prompt in order to build the corpus for the scoring system.

The three modules in AEA are NLP, dimensionality reduction components and grade definition. The NLP module consists of the morphological analyser and a Constraint Grammar Parser for the Finnish language (FINCG). In this module, words are reduced to their base forms in a process called lemmatisation and their part-of-speech tags are obtained. Then stop word removal is performed, followed by the construction of a word-by-context (WCM) matrix for every word in each corpus document. Then, entropy-based term weighting is applied to the matrix.

In the dimensionality reduction component, using the SVD technique of LSA, the matrix is processed in such a way to restore only the underlying semantic structure and to eliminate the other details, thereby reducing the dimensions of the matrix.

Finally, in the grade definition phase, the grade of a new essay is assigned. The semantic similarity between the document vector of the essay and the document vectors of each of the corpus document is determined by calculating the cosine between them. Then, depending on the similarity value and the limits of grade categories, a single grade category is assigned to the essay.

Success: The evaluation experiments used 100-150 essays, divided into three essay datasets on topics of education, marketing and software engineering. A Spearman correlation of 75% between the grades assigned by the system and human graders was reported.

Shortcomings:

1. It is based on the ‘bag of words’ approach.
2. This system can be used to grade essays in the Finnish language only.
3. It is prompt-specific and hence, needs to be trained specifically for every prompt.

4. It can only be used for essays which are written based on course materials. The authors report that their system performed very poorly for open-ended prompts. Hence, it cannot be used for essays of narrative genre [16].
5. The developers of this system themselves state that AEA considers only the content of the essay and ignores all aspects of spelling and syntax in the essay [54]. Hence, it cannot be used for analytic scoring to obtain the scores of each criterion individually.

#### 2.6.2.3. Jess

In the AEG systems mentioned above, the system has been designed to replicate the scores produced by human raters. Hence, it can be said that these scoring systems are based on ratings by expert raters. However, a recent AEG system called 'Jess' is based on expert writings instead of expert ratings and is an AEG system for the Japanese language. From professional writing in Japanese newspapers, it extracts various feature values to develop the system [56]. The various features are converted into three main features: rhetoric, organization and content which Jess aims to evaluate in essay grading. In order to measure the rhetoric, the ease of reading, diversity of the vocabulary, percentage of large words, etc. are calculated. To extract the feature value of organization, the E-rater method of looking for certain phrases and cue words is adopted. Several words depicting relationships between phrases are taken into account and using the probability of a forward connection or a backward connection, the organization in the essay is assessed. To measure the content in the essay, Jess employs the LSA technique. Overall, the Jess model is largely based on the E-rater method.

At the time of writing this thesis, Jess can be executed over the web at http://coca.rd.dnc.ac.jp/jess/.

Success: In an experiment using 143 essays written by university students on the topic 'Festivals in Japan', there was 84% correlation between the scores assigned by Jess and those as-

signed by human graders. This value was higher than the inter-rater correlation of 73% between human raters.

When Jess was evaluated on essays previously scored by E-rater, English essays were converted to Japanese and then processed. In this test, Jess reports a correlation of 0.83 with the average of expert rater's scores. The authors report that this is larger when compared to the correlation coefficient between expert rater's cores (0.70). Further, authors report that this method performed better than E-rater for essays which were shorter in length but had the same writing format.

Shortcomings:

1. Jess is suitable only for the Japanese language.
2. The authors of this system report that Jess is not suitable for essays that involve scientific and technological language [56].
3. It has been reported that it assigns a low value for the 'content' of the essay, even though the essay responded well to the question prompt.
4. There is not enough evidence that Jess can be used for grading narrative essays.

#### 2.6.2.4. MarkIT

MarkIT is a hybrid method combining shallow NLP techniques and multiple linear regression, which is a statistical technique. It is designed to automatically grade student essays against a model answer provided by the teacher. From the model answer, concepts are extracted.

The student response is initially pre-processed by a stemming algorithm and then stop word removal is performed. Then a "chunker" is used to obtain chunks of text. For every word in the phrases, its respective concept number is obtained from the Macquarie thesaurus and

phrases are represented as numbers [57]. In this way, propriety representation of the meaning and content of the essay is built. Finally, the cosine value for the angle between the two vectors is obtained in order to determine the similarity between the vectors and accordingly, a grade is assigned [58].

MarkIT uses the statistical technique of multiple linear regression in order to assign a grade to the student essay. An essay dataset of around 50-200 essays is used to develop the regression equation.

The average word length and the number of NP adjectives are some of the significant independent variables in the regression equation, as reported by [57].

Success: In an evaluation trial involving 20 law essays, 72% correlation between human and computer scores was reported [59]. MarkIT can automatically generate visual as well as textual feedback [59]. The developers of this system claim that it can process a 400-word essay in about 3 seconds [60]. It can be run on a standard Windows PC and is highly portable [61]. The developers assert that the automated technique used in MarkIT is robust enough to quickly and formally represent a copious amount of free text with the help of semantic representations [59].

Shortcomings:

1. MarkIT is capable of grading only the content of the essays.
2. It is the only AEG system that has been applied to grading essays according to the NAPLAN rubric. However, MarkIT only assesses the overall content and leaves the important task of final assignment of scores to the human graders.

3. This system is based on the "bag of words" approach. This means that the two sentences "The policeman caught the thief" and "The thief caught the policeman" will be treated as the same, although the second sentence is illogical.
4. In order to achieve a high score, students need to demonstrate the set concepts stipulated in the model answer. Accordingly, an exceptional student response containing concepts derived from other material would be assigned a low score.

#### 2.6.2.5. Using Generalised LSA (G-LSA)

[62] propose an AES system using the generalised LSA technique. In this method, some preprocessing steps are carried out on an essay dataset and then with a slight modification to the LSA method (described earlier in this section), better performance than the LSA method is demonstrated. The essay dataset used for the development of this method was initially prescored by human graders. Then, stop words are removed and stemming is carried out after which a n-gram by document matrix is created using the frequency of n-grams in the document. Finally, this matrix is decomposed using the SVD technique and then depending on the cosine correlation value, the essay is assigned a suitable grade.

Success: This system was trained by using a dataset of 960 essays and then tested on a dataset of 120 essays. An accuracy range of 89% to 96% is reported during the evaluation of the system.

Shortcomings:

1. It appears that except for the content of the student answer, other features of the answer such as sentence structure, etc. are not taken into account.
2. This system is suitable for holistic scoring but not for analytic scoring.
3. This system assumes that the essays to be graded are all properly constructed responses, which is not necessarily true.

4. There is not enough evidence that this system can be used to grade narrative essays.

### 2.6.3. Text Categorisation Techniques (TCT)-based methods

TCT is an AEG system that uses several text classification techniques to perform automated essay grading system. One or more of these techniques, with little variation/modification, is used by other AEG systems to perform the task of automated grading of essays. In this section, the working of AEG systems that are based on TCT-based methods is explained.

#### 2.6.3.1. Text Categorization Techniques (TCT)

Text Categorization Techniques (TCT) was proposed by Larkey at the University of Massachusetts in USA. The idea behind developing TCT was ‘to train binary classifiers to distinguish ‘good’ from ‘bad’ essays, and use the scores output by the classifiers to rank essays and assign grades to them’ ([20], p.90). A Bayesian classification approach, originally proposed by [63], is used to distinguish ‘good’ essays from ‘bad’ essays. Firstly, text is pre-processed by stop word removal and stemming. Then, feature selection is performed where the most representative features of the text are identified. Then, the essay dataset is divided at various score points and finally, binary classifiers are trained to distinguish between essays of score '4' from essays of scores ‘1’, ‘2’ and ‘3’ and so on.

A further two techniques are used. The *k* most similar essays to the student essay are identified using the Inquery Retrieval system and then the student essay is assigned a score that is assigned to the *k* most similar essays [52].

Finally, eleven text complexity features are automatically extracted from the text. Some of these features are the number of characters in the essay, the average word length, average sentence length, number of different words in the essay and so on [20]. All these features are used either on their own or in conjunction with the previous two methods, in a step-wise line-

ar regression equation, to obtain the essay score. According to Larkey, Bayesian classifiers performed better than the other two techniques [20]. An important contribution of the developers of this method is finding the 'exact agreement' between the grades assigned by the human grader and the computer, that is, the percentage of times that the system and human grader scored exactly the same value.

Success: An evaluation of this system was carried out on about 40 essays in each of the subjects of social studies, physics and law. Consequently, an exact agreement of 60% was reported and an adjacent agreement of 100% was reported when all the three criteria were used. When this system was applied to essays on general opinion, 55% exact agreement and 97% adjacent agreement was reported. Furthermore, a correlation of 88% was reported for the general opinion essays.

Shortcomings:

1. This method has not been applied to narrative essays. Hence, there is no evidence that it can be used to grade narrative essays.
2. It provides holistic scores. Hence, it cannot be used to assign analytic scores individually to criteria such as spelling, vocabulary and so on.
3. It makes an assumption that the essays to be scored reflect a properly constructed response, which is not necessarily true.
4. It assumes that the relationship between the essay features and its grade is linear, which is not necessarily true.

#### 2.6.3.2. Bayesian Essay Test Scoring sYstem (BETSY)

The Bayesian Essay Test Scoring sYstem (BETSY), a program for essay classification, uses Bayesian text classification techniques [64]. It was developed by Lawrence M. Rudner at the University of Maryland, College Park, USA. The Bayesian Computer Adaptive Testing

(CAT) developed by [65, 66] is used and was further enhanced by Lawrence M. Rudner to develop BETSY.

In BETSY, both the content and style of the essay are taken into consideration [1]. Initially a large collection of essays is graded by expert human raters and these essays are used to determine conditional probabilities of presence of features. Initially, equal prior probabilities are assumed for each feature. After examining each feature in an essay, the probability class (appropriate, partial, inappropriate) for that feature is updated using Bayes theorem. The updated values are called posterior probabilities and are treated as new prior probabilities. This process is repeated until all the features have been taken into consideration. The score with the highest posterior probability is assigned to the essay [67].

For the pre-processing of essays, BETSY performs stemming and stop word removal. For feature selection, the entropy reduction method is used and the features which demonstrate maximum potential information gain are chosen as the best indicators [1,64].

Using two naïve Bayes models, the most likely classification of the essay into a four-point scale (extensive, essential, partial, unsatisfactory) is obtained. In the Multivariate Bernoulli Model (MBM), each essay is considered a special case of all the calibrated features. The probability of each score for a given essay is computed by multiplying the probabilities of features present in the essay, including the features that are not present in the essay. The conditional probability of the absence of a feature is 1 minus the probability of the feature presence. The number of times a feature appears in the essay is not taken into consideration in this method. According to [67], this method is suited for a task that has a fixed number of attributes/features.

In the Multinomial Model, each essay is considered to be a sample of all the calibrated features. The probability of each score for a given essay is computed as the product of probabili-

ties of the features present in the essay. The multiple uses of a feature in the essay are considered in this model. In both methods, Laplacian correction is used which means that the frequencies are seeded with 1 to prevent zero probabilities which would render all the other features useless [64].

After calibration of the two models, new essays are scored using them. It is reported that the multinomial model is computationally faster as only the features of the essay need to be examined whereas in the multivariate model, all the features in the vocabulary need to be examined [67]. Further, the multinomial model performs better with large vocabulary sizes whereas MBM performs better with small vocabulary sizes.

Success: [64] claim that BETSY works on an approach that may incorporate the best features of PEG, LSA, and E-rater, plus it is simple to implement and can be used for a wide range of content areas. Further, it is easy to explain to non-statisticians.

Rudner and Liang conducted experiments using the two Bayes models that were calibrated using 482 essays with two score points. The calibrated models were then applied to 80 new pre-scored essays, with 40 in each score group. The authors achieved an accuracy of 80% using BM and an accuracy of 74% using MBM.

Shortcomings:

1. It is computationally very intensive.

2. It is based on Bayesian text classification techniques which is a “Bag of words” approach. This means that word order is not considered at all.

3. It can only be applied on short essays.

4. It requires a large sample of training essays, typically hundreds or even thousands of them.

5. It can be used only for essay classification and not for assigning numerical scores to essays.

6. It does not regard that test essays can contain improperly constructed responses which is too optimistic.

#### 2.6.3.3. CarmelTC

CarmelTC is a free text assessment module, incorporated into the virtual learning environment system, Carmel. Similar to TCT, CarmelTC employs text classification techniques and Naïve Bayes classification.

The methodology of CarmelTC is: firstly, the student response is split into sentences, then the Bayesian technique is used to find the probability of the presence of the correct feature that represents each sentence. Then, a vector is generated depending on the presence or absence of each correct feature and finally, the rules for identifying sentence classes based on these feature vectors are incorporated with the ID3 tree learning algorithm (as cited in [2]). The rule-based learning in text categorisation techniques is coupled with the syntactic functional analyses of text to extract information about the features of the text [68]. Further, it overcomes the shortcoming of the ‘bag of words’ approach by considering word order. This system can be used to assess essays where causal relations are considered. This is a significant contribution of this system, which would be beyond the scope of other 'bag of words' approaches.

Success: This system was tested with 126 physics essays and results reported were 90% precision, 80% recall and 8% false alarm rate.

Shortcomings:

1. There is not enough evidence that this system can be used to grade narrative essays.
2. This system does not regard improperly constructed essays in the essays to be graded.

3. This system assumes that the relationship between the essay features and the grade is linear, which is not necessarily true.

#### 2.6.3.4. Using the Nearest Neighbour algorithm and information retrieval

A web-based AEG system, developed to mark Malay essays on history, uses the nearest neighbour technique from within the field of information retrieval [69]. The main idea behind using the nearest neighbour technique in essay grading is to find the closest model answer to the students' answer and return a score with the minimum distance.

There are four modules in this system, one for indexing, structuring of the model answer, matching and mark processing. In 'indexing', the document is pre-processed by removing hyphens, commas and full stops; the conversion of uppercase letters to lower case; stop word removal and stemming. Then, using a set of representative keywords from the document, it is indexed and organized to allow effective keyword searching. In this way, the student's essay is query processed to be an indexed essay. Likewise, the model answer is indexed to be a model answer scheme. In the mark processing module, the nearest- neighbour algorithm is employed with the overlap metric. By an effective comparison of keywords between the model answer and the student answer, the overlap between them is calculated. The marking scheme provided by the teacher determines the marks for each answer. Accordingly, the AEG system assigns the marks to the essay.

Success: The test set comprised six questions with ten student answers for each. For every question, five different paraphrases of the model answer were used. An accuracy of about 91% is reported in this evaluation.

Shortcomings:

1. This system is suitable for essays in the Malay language and has been applied only to the History domain.
2. It is susceptible to spelling mistakes in the main keywords. If the student's response contains the important keywords but has spelt them incorrectly, then the keywords will not be detected by their system and hence the student will not receive the appropriate score.
3. It does not analyse the grammar and sentence structure of the responses.

#### 2.6.3.5. Using KNN algorithm

An AEG system for grading Chinese essays is built using the K-nearest neighbour algorithm [70]. In this system, essays are first transformed and their vector space model (VSM) is constructed. This vector space model consists of words, phrases and arguments as the features and every vector is assigned a value by two methods: term frequency and inversed document frequency weight. The similarity between the VSM of the test essay and the training essays is computed using a cosine formula. Then, the K-nearest neighbours of the test essay are identified and the respective essay score is assigned to the test essay. An accuracy of 76% is reported in an evaluation of this method.

Shortcomings:

1. This system is only suitable for Chinese essays.
2. It is based on the 'bag of words' approach.
3. It can only be used for holistic scoring.

### 2.6.4. Miscellaneous techniques in Automated Essay Scoring

In this section, the working of AEG systems based on miscellaneous techniques is explained.

#### 2.6.4.1. Using connections between paragraphs

Researchers have proposed a Chinese AES system based on the connections between various concepts in paragraphs [71]. The basic assumption is that a paragraph consists of concepts and sub-concepts. The concepts and sub-concepts share a concept-level hierarchy. Furthermore, various concepts appear in a certain order in an essay, depicted by R-chains. The authors suggest that a test essay can be scored depending on the similarity of its concept hierarchy and R-chain with that of the pre-scored essays.

Success: This method has an accuracy rate of 84%.

Shortcomings:

1. The exact rate of this method was only 37%.
2. This method works only for essays when the dataset consists of at least 200 essays.
3. This system can be used only for holistic scoring but not to assign individual scores to criteria such as spelling and vocabulary.

#### 2.6.4.2. Using a set of literary sememes

[72] propose an AEG method that is based on the number of literary sememes in an essay. Sememes are words that are semantically related to a concept. For example, the concept 'school' has sememes 'place', 'education', 'learning' and 'teaching'. In this method, the total number of sememes for the concepts in an essay is determined and then the essay score is assigned according to the scale given in Table 2.1.

**Table 2.1: Essay scores associated with the number of sememes present in the essay**

| Number of sememes in essay | Essay Score |
|---|---|

| | |
|---|---|
| Less than 2 | 1 |
| 3-6 | 2 |
| 7-13 | 3 |
| 14-23 | 4 |
| 24-43 | 5 |
| More than 43 | 6 |

Success: Experiments have found that as the number of literary sememes increases, the essay score increases. This method of AES shows an exact rate of 44.2% and an accuracy rate of 91.6%.

Shortcomings:

1. This method has been tested for essays on only one topic ‘Recess at School’ and only for one grade level (eighth). Hence, there is insufficient evidence that this system can be used to grade narrative essays.
2. This method does not model the non-linear nature of the relationship between essay features and its grade. In fact, it assesses only the content of the essay.
3. It can be used for assigning only holistic scores in the range of 0 to 6. Hence, it cannot be used for analytic scoring.
4. This system assumes that the essays to be graded are properly constructed responses, which is not necessarily the case.

#### 2.6.4.3. Using unsupervised learning based on a voting algorithm

[73] propose an AES system that uses a small set of unscored essays on the same topic. An unsupervised learning model based on a voting algorithm is used to classify essays into dif-

ferent clusters. The underlying premise is that "the voting essays are prone to attract the essays that are similar to them and that will lead to essay clustering". Six clusters are used, one for each score point. Initial weights are obtained by allowing the model to iteratively learn the feature information from the essays (henceforth 'voting essays'). Then, a new target essay obtains votes from all the voting essays and a z-score, then finally a score is calculated for the target essay. Depending on the score, the target essay is classified into one of the clusters and is graded as per the other essays in the cluster.

Success: This system is reported to have yielded an exact agreement of 52% and an adjacent agreement of approximately 94%. The highlight of this system is that it does not need pre-graded essays. Furthermore, it is claimed that this system can detect gibberish essays (which contain a large number of meaningless terms) and off-topic essays.

Shortcomings: Many significant shortcomings can be identified in this system.

1. Although the authors claim that it can be adapted to other languages, currently it is suitable only for grading essays in the Chinese language.
2. It uses a 'bag-of-words' approach.
3. Like PEG, it employs indirect features rather than direct measures of sentence variety and argument analysis.
4. The authors caution that this system cannot be applied to essays in the narrative writing genre.
5. This system can only be used for holistic scoring where the essay score ranges from score 1 to 6.

#### 2.6.4.4. Using a modified BLEU algorithm

An AEG system for assessing free-text answers is proposed with a few modifications to the original BLEU algorithm (earlier used for machine translation). It can be used to compare a

candidate free-text answer with a set of expert reference answers [74]. For every question on e-Learning, several reference answers are obtained from tutors and expert instructors. All these reference answers are assigned M-BLEU scores independently by taking the n-gram counts (number of n-grams in the reference answers where n-gram is the sequence of words that appear consecutively in a text) and the word weights into account. The student answer is compared against the reference answer that has the maximum M-BLEU score. Depending on the level of similarity between the student answer and the reference answer, the appropriate score is assigned to the student answer. Hence, the more similar the student answer is to the reference answer, the more marks it will score.

Success: The authors claim that this automatic assessment method using the M-BLEU algorithm has shown a maximum Pearson correlation of 85% and an adjacent agreement of 0.75 with the experts.

Shortcomings:

1. This method allows the students to use a spell-check feature when writing their scripts, in order to eliminate misspelled words so as to avoid mismatches when matching the student answer against the reference answer. As a result, students will not bother to learn the correct spelling themselves.
2. Additionally, it appears that except for the content of the student answer, other features of the answer such as sentence structure, etc. are not taken into account.
3. This system is suitable for holistic scoring but not for analytic scoring.
4. This system assumes that the essays to be graded are all properly constructed responses, which is not necessarily true.
5. This system is prompt-specific, hence it has to be retrained for every new prompt.

In the next section, a critical review of the AEG systems from all four categories is presented.

## 2.7. Critical review of AEG systems

This section presents a critical review of the existing AEG methods which have been explained in the previous sections of this chapter. The purpose of performing a critical evaluation of the existing systems is to find the main issues that have not been addressed so far in the literature. Therefore, the issues that will be addressed in this thesis are highlighted in the following questions:

- Q1: Can the AEG system be used to grade essays in English language?
- Q2: Can the AEG system be used for analytic scoring?
- Q3: Can the AEG system be used for scoring narrative essays?
- Q4: Does the AEG system model both the linear and non-linear relationships between the essay features and the essay grade?
- Q5: Can the AEG system be trained and calibrated using a relatively small dataset (preferably less than 200 essays)?
- Q6: Is the AEG system computationally non-intensive?
- Q7: Can the AEG system handle improperly constructed responses?

Each of the AEG systems explained so far will be assessed against these seven questions in order to carry out the critical review. As seen in the previous sections, the existing AEG systems have been developed for a variety of languages. However, our interest lies in an AEG system that is capable of grading essays in the English language, which is covered by Q1.

Further, we need to evaluate if the AEG system can be used for analytic or multi-trait scoring, which is covered in Q2. Some existing AEG systems can be used for holistic as well as analytic scoring whereas other systems can be used for holistic scoring only. However, some

AEG systems do not explicitly mention the scoring method for which the system can be used. Hence, for these essays, it is assumed that they can be used for grading holistic scoring only because it is the conventional and most common method of essay grading, as mentioned previously.

Once the scoring method is evaluated, we want to investigate if the AEG method can be used for scoring narrative essays, which is covered by Q3. Some AEG systems explicitly mention the genres that can be assessed using the system. Where it is not mentioned explicitly, we assume that the AEG system cannot be used to grade narrative essays because this is the most challenging and least common type of genre to assess.

Further, we need to evaluate if the AEG system models both the linear and non-linear relationships between the essay features and the essay grade, which is covered by Q4. It is important to model both the nature of relationships because otherwise essay scoring might not be fair in some cases. For example, most AEG systems assign the score depending on the length of the essay. If the essay is long, then it is assigned a higher grade than otherwise. However, this might not be the case because although the essay is long, it might contain a lot of irrelevant terms which do not contribute much to the content of the essay. Hence, it is desirable that the AEG system is able to model both the nature of relationships between the essay feature and the grade.

Another highlight of an ideal AEG system is that the system should require a relatively small dataset for the training and calibration of the grading model, which is covered by Q5. To answer this question, we consider a dataset of less than 200 essays as relatively small, which is a widely held view in the field of AEG.

Further, it is important that the AEG system is not computationally intensive, which is covered by Q6. If the system is computationally intensive and resource hungry, then it will re-

quire specialised hardware and software and will take considerable time to produce results. In the worst cases, such as for PS-ME, it cannot be used in real time due to such heavy processing requirements. Hence, an ideal AEG system should be computationally non-intensive.

Finally, since the AEG system deals with free text written mostly by students, it is essential that the system has a methodology to detect and flag improperly constructed responses, which is covered by Q7. An ideal AEG system should be able to not only detect but also handle such responses because they can potentially run down the performance of the AEG system.

The assessment of AEG systems against each of these questions is shown in Table 2.2.

To the best of our knowledge, at the time of writing this thesis, there are 24 AEG systems of which 10 systems are hybrid, 5 systems are LSA-based, 5 are TCT-based and the remaining 4 systems are miscellaneous technique-based.

**Table 10.2: Critical analysis of the existing AEG systems**

| No. | AEG system | Critical Evaluation Questions | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 |
| 1. | PEG | ✓ | × | × | × | × | × | × |
| 2. | E-rater | ✓ | × | × | × | × | × | ✓ |
| 3. | E-rater V.2 | ✓ | × | × | × | × | × | ✓ |
| 4. | Criterion | ✓ | × | × | × | × | × | ✓ |
| 5. | SEAR | ✓ | × | × | × | ✓ | ✓ | × |
| 6. | IEMS | ✓ | × | × | ✓ | ✓ | ✓ | × |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7. | PS-ME | ✓ | × | × | ✓ | ✓ | × | × |
| 8. | Intellimetric | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ |
| 9. | My!Access | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ |
| 10. | [43] | × | × | × | × | ✓ | ✓ | × |
| 11. | IEA | ✓ | × | × | × | ✓ | × | × |
| 12. | AEA | × | × | × | × | ✓ | × | × |
| 13. | Jess | × | ✓ | × | × | ✓ | × | × |
| 14. | MarkIT | ✓ | × | ✓ | × | ✓ | ✓ | × |
| 15. | Using G-LSA | ✓ | × | × | × | × | × | × |
| 16. | TCT | ✓ | × | × | × | ✓ | × | × |
| 17. | BETSY | ✓ | × | × | × | × | × | × |
| 18. | CarmelTC | ✓ | ✓ | × | × | ✓ | × | × |
| 19. | [70] | × | × | × | × | ✓ | × | × |
| 20. | [69] | × | × | × | × | ✓ | × | × |
| 21. | [71] | × | × | × | × | × | ✓ | × |
| 22. | [72] | ✓ | × | × | × | ✓ | ✓ | × |
| 23. | [73] | × | × | × | ✓ | ✓ | ✓ | ✓ |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 24. | [74] | × | × | × | × | ✓ | × | × |
| Total | | 16 | 4 | 3 | 5 | 17 | 7 | 6 |

As shown in Table 2.2, of the 24 existing AEG systems, only 16 can be used for grading essays in English language. The other 8 systems comprise 5 systems for grading essays in Chinese and one system for grading essays in the Japanese, Malay and Finnish languages. In the evaluation of Q2, only 4 AEG systems can be used for analytic scoring whereas most of the existing systems, 20 to be precise, can be used for holistic scoring. Similarly, in the evaluation of Q3, only 3 systems can be used for scoring narrative essays whereas a majority of 21 existing systems cannot be used for scoring essays of the narrative genre.

Of the 24 available AEG systems, only 5 systems model both the linear and the non-linear nature of relationships between the essay features and the essay grade, whereas 19 systems assume that the relationships are linear. The next evaluation Q5 checks if the system can be trained using a dataset of less than 200 essays. The results for Q5 in Table 2.2 demonstrate that 17 existing systems satisfy this evaluation question, whereas 7 systems do not satisfy it because they need at least 200 essays, with BETSY requiring anywhere between hundreds to even up to thousands of essays as the training set. Of the existing AEG systems, a total of 7 systems are computationally non-intensive whereas 17 systems are resource hungry and require huge processing requirements, PS-ME being the worst in this case, as it cannot be used for grading essays in real time for the same reason. The next evaluation Q7 checks if the system can handle improperly constructed responses. The results for this evaluation are shown in column Q7 in Table 2.2. A total of only 6 AEG systems can handle some improperly constructed responses whereas 18 systems are not equipped with any methodology to handle such responses. Of the 6 systems which can handle such responses, most of them can handle only

one or two types whereas one system can handle a maximum of three different types of anomalous responses.

In light of the above evaluation, it is realised that not one AEG system satisfies all the seven evaluation questions. Hence, the issues that are raised as a result of this evaluation are as follows.

1. The AEG system should be able to grade essays in the English language.
2. It should be able to perform analytic scoring such that scores for criteria such as spelling and vocabulary can be individually assigned.
3. It should be able to score essays of the narrative genre.
4. It should be able to model both the linear as well as non-linear nature of relationships between the essay features and the essay grade.
5. It should require a relatively small dataset for training and calibration in order to build the scoring model.
6. It should avoid intensive and resource-hungry computations and be able to perform best with maximum use of available resources.
7. It should be able to handle improperly constructed responses thereby avoiding their negative effect on the overall performance of the system.

The research issue and objectives will be coined keeping the above issues in perspective. In the next section, the main points of this chapter are presented and the chapter is concluded.

## 2.8. Conclusion

In this chapter, the various automated systems such as ETS1, C-rater, ATM and Automark for marking short answer type responses were described, after which the working of semi-automated essay grading systems such as MARC, Markin32, Student Essay Viewer and ePen was outlined. It is important to note that semi-automated essay grading systems provide only a visual interface to display the essay and some annotation buttons for the purpose of assisting the human marker in the essay marking process. However, the onus of marking the essay lies with the human marker in these types of systems. On the other hand, automated essay grading systems are capable of carrying out the marking process without human intervention.

The available AEG systems into were broadly classified into four categories: hybrid systems, LSA-technique based systems, TCT-technique based systems and miscellaneous techniques-based systems. The working of several available AEG systems in each category was explained: Hybrid systems such as PEG, E-rater, E-rater V.2, Criterion, SEAR, IEMS, Intellimetric, My!Access and an AES system for grading CET4 essays; LSA-technique-based methods such as IEA, AEA, Jess, MarkIT and a method using G-LSA; TCT-techniques-based methods such as TCT, BETSY, CarmelTC, kNN-based method for grading Malay essays and another for grading Chinese essays; Miscellaneous techniques-based methods such as using connections between paragraphs, using set of literary sememes, using unsupervised learning based on a voting algorithm and using the modified BLEU algorithm. Finally, a critical evaluation all the available systems with regard to the seven evaluation criteria was given. It was found that none of the available systems satisfied the seven criteria; hence there is a need for an AEG system which can satisfy all the criteria to perform automated essay grading using the analytic scoring method for narrative essays.

## 2.9. References


[1] S. Dikli, "An overview of automated scoring of essays," *The Journal of Technology, Learning, and Assessment*, vol. 5(1), 2006.

[2] D. Perez-Marin, I. Pascual-Nieto, and P. Rodriguez, "Computer-assisted assessment of free-text answers," *The Knowledge Engineering Review*, vol. 24(4), pp. 353–374, 2009.

[3] (2008, 13 Sep 2011). *IR-Book*. Available: http://nlp.stanford.edu/IR-book/html/htmledition/contents-1.html

[4] D. Marcu, *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, MA: The MIT Press, 2000.

[5] M. D. Shermis and J. Burstein, *Automated Essay Scoring: A cross disciplinary perspective*. Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[6] D. Whittington and H. Hunt, "Approaches to the computerized assessment of free text responses," presented at the Sixth International Computer Assisted Assessment Conference, Loughborough University, UK, 1999.

[7] J. Burstein, C. Leacock, and R. Swartz, "Automated evaluation of essay and short answers," presented at the Sixth International Computer Assisted Assessment Conference, Loughborough,UK, 2001.

[8] S. Valenti, F. Neri, and A. Cucchiarelli, "An overview of current research on automated essay grading," *Journal of Information Technology Education*, vol. 2, pp. 319-330, 2003.

[9] D. Callear, I. Jerrams-Smith, and V. Soh, "CAA of Short NonMCQ Answers," in *Proceedings ofthe fifth International Computer Assisted Assessment Conference*, Loughborough, UK, 2001.

[10] D. Perez-Marin, I. Pascual-Nieto, and P. Rodriguez, "Computer-assisted assessment of free-text answers," *The Knowledge Engineering Review*, vol. 24(4), pp. 353-374.

[11] J. Cowie and W. Lehnert, "Information Extraction," *Communications of the ACM*, vol. 39(1), pp. 80-91, 1996.

[12] T. Mitchell, T. Russel, P. Broomhead, and N. Aldridge, "Towards robust computerized marking of free-text responses," presented at the Sixth International Computer Assisted Assessment Conference, Loughborough University, Loughborough,UK, 2002.

[13] S. Marshall and C. Barron, "Marc-methodical assessment of reports by computer," *System*, vol. 15(2), pp. 161–167, 1987.

[14] J. Burston, "Computer-mediated feedback in composition correction," *CALICO Journal*, vol. 19(1), pp. 37-50, 2001.

[15] E. Moreale and M. Vargas-Vera, "Genre analysis and the automated extraction of arguments from student essays," in *Proceedings of the Seventh International Computer Assisted Assessment Conference*, Loughborough University, Loughborough, 2003.

[16] T. Kakkonen, N. Myller, E. Suttinen, and J. Timonen, "Comparison of Dimension Reduction Methods for Automated Essay Grading," *Educational Technology & Society*, vol. 11(3), pp. 275-288, 2008.

[17] R. Johnson, *Elementary Statistics*, 5 ed.: PWS-Kent Publishers, 1988.

[18] Y. Yang, "An evaluation of statistical approaches to text categorization," Carnegie Mellon University, CMU-CS-97-127, 1997.

[19] P. Foltz, "Latent semantic analysis for text-based research," *Behavior Research Methods,* vol. 28, pp. 197-202, 1996.

[20] L. S. Larkey, "Automatic essay grading using text categorization techniques," in *Proceedings of the 21st ACM/SIGIR (SIGIR-98),* Melbourne, Australia, 1998, pp. 90-96.

[21] E. Page, "The imminence of grading essays by computer," *Phi Delta Kappan,* vol. 47, pp. 238-243, 1966.

[22] D. P. Marin, "Automatic evaluation of users' short essays by using statistical and shallow natural language processing techniques," Master's thesis, Universidad Autónoma de Madrid, Madrid, 2004.

[23] A. Ben-Simon and R. E. Bennett, "Toward More Substantively Meaningful Automated Essay Scoring," *Journal of Technology, Learning, and Assessment,* vol. 6(1), 2007.

[24] E. Cotos and N. Pendar, "Automated diagnostic writing tests: Why? How?," in *Towards Adaptive CALL: Natural Language Processing for Diagnostic Language Assessment,* Ames, Iowa, 2008.

[25] J. Burstein, K. Kukich, S. Wolff, L. Chi, and C. M, "Enriching automated essay scoring using discourse marking," presented at the Workshop on Discourse Relations and Discourse Marking, Annual Meeting of the Association of Computational Linguistics, Montreal, Canada, 1998.

[26] J. Burstein, K. Kukich, S. Wolff, C. Lu, M. Chodorow, L. Braden-Harder, and M. D. Harris, "Automated scoring using a hybrid feature identification technique," presented at the Annual Meeting of the Association of Computational Linguistics, Morristown, NJ, 1998.

[27] J. Burstein and D. Marcu, "Toward Using Text Summarization for Essay-Based Feedback," presented at the Septième Conference Annuelle sur Le Traitement Automatique des Langues Naturelles (TALN), Lausanne, Switzerland, 2000.

[28] J. Cheville, "Automated scoring technologies and the rising influence of error," *The English Journal*, vol. 93(4), pp. 47-52, 2004.

[29] D. E. Powers, J. C. Burstein, M. Chodorow, M. E. Fowles, and K. Kukich, "Stumping E-Rater: challenging the validity of automated essay scoring," Educational Testing Service, Princeton, NJ ETS RR-01-03 and GREB-98-08bP, 2001.

[30] H. W. Lam, T. Dillon, and E. Chang, "Determining Writing Genre: Towards a Rubric-based Approach to Automated Essay Grading," presented at the IEEE International Conference on Advanced Information Networking and Applications (AINA), Singapore, 2011.

[31] S. Darus, S. Stapa, and S. Hussin, "Experimenting a Computer-Based Essay Marking System at Universiti Kebangsaan Malaysia," *Jurnal Teknologi*, vol. 39(E), pp. 1-18, 2003.

[32] J. R. Christie, "Automated essay marking-for both style and content," presented at the Third Annual Computer Assisted Assessment Conference, Loughborough,UK, 1999.

[33] P. Y. Ming, A. A. Mikhailov, and T. L. Kuan, "Intelligent Essay Marking System," *Learners Together*, 2000.

[34] N. K. Nikitas, "Computer Assisted Assessment (CAA) of Free-Text: Literature Review and the Specification of an Alternative CAA System," 2010, pp. 116-118.

[35] B. S. Bloom, *Taxonomy of educational objectives: The classification of educational goals (1st ed.)*: Harlow, Essex, England: Longman Group, 1956.

[36] O. Mason and I.-G. Stephenson, "Automated free text marking with Paperless School," presented at the Sixth International Computer Assisted Assessment Conference, Loughborough,UK, 2002.

[37] S. Elliot, "IntelliMetric: from here to validity," in *Automated essay scoring: a cross disciplinary approach*, M. D. Shermis and J. C. Burstein, Eds., ed Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[38] L. M. Rudner, V. Garcia, and C. Welch, "An Evaluation of the IntelliMetric[SM] Essay Scoring System," *Journal of Technology, Learning, and Assessment*, vol. 4(4), 2006.

[39] V. Learning. (2011). Available: http://www.vantagelearning.com/

[40] J. Wang and M. S. Brown, "Automated Essay Scoring Versus Human Scoring:A Comparative Study," *Journal of Technology, Learning, and Assessment*, vol. 6(2), 2007.

[41] B. Davies and T. Gralton, "Trial of Automated Essay Scoring:new directions for national assessment in Australia," presented at the International Association for Educational Assessment, Brisbane,Australia, 2009.

[42] C.-F. E. Chen and W.-Y. E. Cheng, "Beyond the Design of Automated Writing Evaluation: Pedagogical Practices and Perceived Learning Effectiveness in EFL Writing Classes," *Language Learning & Technology*, vol. 12(2), pp. 94-112, 2008.

[43] L. Yali, "Automated Essay Scoring System for CET4," in *Second International Workshop on Education Technology and Computer Science*, 2010, pp. 94-97.

[44] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and H. R. A., "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391-407, 1990.

[45] K. W. K. Chung and H. F. O'Neil, "Methodological approaches to online scoring of essays," University of California, Los Angeles, Technical Report no. 461, 1997.

[46] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychological Review,* vol. 104, pp. 211-240, 1997.

[47] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science,* vol. 41(6), pp. 391-407, 1990.

[48] T. K. Landauer, D. Laham, B. Rehder, and M. E. Schreiner, "How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans," in *19th Annual Conference of the Cognitive Science Society,* Mahwah, NJ, 1997, pp. 412–417.

[49] P. W. Foltz, D. Laham, and T. K. Landauer. (1999, 3/08/2011). The Intelligent Essay Assessor: Applications to Educational Technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning 1(2).* Available: http://imej.wfu.edu/articles/1999/2/04/

[50] M. A. Hearst, "The debate on automated essay grading," *Intelligent Systems and their Applications, IEEE,* vol. 15, pp. 22-37, 2000.

[51] P. W. Foltz, D. Laham, and T. K. Landauer, "Automated Essay Scoring: Applications to educational technology," presented at the EdMedia, 1999.

[52] R. Williams, "Automated essay grading: An evaluation of four conceptual models," presented at the 10th Annual Teaching and Learning Forum, Curtin University of Technology, Perth, 2001.

[53] T. Kakkonen, N. Myller, J. Timonen, and E. Sutinen, "Automatic essay grading with probabilistic latent semantic analysis," presented at the Proceedings of the second workshop on Building Educational Applications Using NLP, Ann Arbor, Michigan, 2005.

[54] T. Kakkonen and E. Sutinen, "Automatic Assessment of the Content of Essays Based on Course Materials," in *Proceedings of International Conference on Information Technology: Research and Education*, London, UK, 2004, pp. 126-130.

[55] T. Kakkonen and E. Sutinen, "Evaluation Criteria for Automatic Essay Assessment Systems - There is much more to it than just the correlation," in *Proceedings of the16th International Conference on Computers in Education*, Taipei, Taiwan, 2008, pp. 111-115.

[56] T. Ishioka and M. Kameda, "Automated Japanese essay scoring system based on articles written by experts," presented at the Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Sydney, Australia, 2006.

[57] R. Williams, "The Power of Normalised Word Vectors for Automatically Grading Essays," *Journal of Issues in Informing Science and Information Technology*, vol. 3, pp. 721-729, 2006.

[58] K. R. Parker, R. Williams, P. S. Nitse, and A. S. M. Tay, "Use of the Normalized Word Vector Approach in Document Classification for an LKMC," *Issues in Informing Science and Information Technology*, vol. 5, pp. 513-524, 2008.

[59] R. Williams and H. Dreher, "Formative assessment visual feedback in computer graded essays," *Journal of Issues in Informing Science and Information Technology*, vol. 2, pp. 23-32, 2005.

[60] R. Williams, "A Computational Effective Document Semantic Representation," in *Inaugural IEEE-IES Digital Ecosystems and Technologies Conference DEST '07*, Cairns, Australia, 2007, pp. 410-415.

[61] R. Williams and H. Dreher, "Telecommunications use in education to provide interactive visual feedback on automatically graded essays," in *Proceedings of*

*International Telecommunications Society Africa-Asia-Australasia Regional Conference*, Perth, Australia, 2005.

[62] M. M. Islam and A. S. M. L. Hoque, "Automated essay scoring using Generalized Latent Semantic Analysis," presented at the 13th International Conference on Computer and Information Technology (ICCIT), 2010.

[63] M. E. Maron, "Automatic Indexing: An experimental Inquiry," *Journal of the Association for Computing Machinery,* vol. 8, pp. 404-417, 1961.

[64] L. M. Rudner and T. Liang, "Automated essay scoring using Bayes' Theorem," *The Journal of Technology, Learning and Assessment,* vol. 1(2), pp. 3-21, 2002.

[65] R. E. Welch and T. Frick, "Computerized adaptive testing in instructional settings," *Educational Training Research and Development,* vol. 41, pp. 47-62, 1993.

[66] D. Madigan, E. Hunt, B. Levidow, and D. Donnell, "Bayesian graphical modeling for intelligent tutoring systems," University of Washington,1995.

[67] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classication," in *AAAI-98,Workshop on Learning for Text Categorization*, 1998, pp. 41-48.

[68] C. Guetl, "Moving towards a Fully Automatic Knowledge Assessment Tool," *International Journal of Engineering Technologies in Learning,* vol. 3(1), 2007.

[69] A. Selamat and K. B. Yee, "Web-Based Automated Essay Marking System for Historical Malay Text Using Nearest-Neighbor Technique," presented at the International Conference on Knowledge Management (ICKM), Charlotte, North Carolina, U.S.A., 2005.

[70] B. Li, J. Lu, J.-M. Yao, and Q.-M. Zhu, "Automated Essay Scoring Using the KNN Algorithm," in *Computer Science and Software Engineering, 2008 International Conference on*, 2008, pp. 735-738.

[71] C. Tao-Hsing and L. Chia-Hoang, "Automatic Chinese Essay Scoring Using Connections between Concepts in Paragraphs," in *Asian Language Processing, 2009. IALP '09. International Conference on*, 2009, pp. 265-268.

[72] C. Tao-Hsing, T. Pei-Yen, L. Chia-Hoang, and T. Hak-Ping, "Automated essay scoring using set of literary sememes," in *International Conference on Natural Language Processing and Knowledge Engineering(NLP-KE '08)*, 2008, pp. 1-5.

[73] C. Yen-Yu, L. Chien-Liang, L. Chia-Hoang, and C. Tao-Hsing, "An Unsupervised Automated Essay Scoring System," *Intelligent Systems, IEEE*, vol. 25, pp. 61-67, 2010.

[74] F. Noorbehbahani and A. A. Kardan, "The automatic assessment of free text answers using a modified BLEU algorithm," *Computers & Education*, vol. 56, pp. 337-345, 2011.

# Chapter 3: Problem Definition

## 3.1. Introduction

In this chapter, the types of most common writing genres are described. Then the problem overview and the research question that this thesis hopes to address are mentioned. With the intention to clearly adumbrate the research problem of this thesis, the nature of National Assessment Program – Literacy and Numeracy (NAPLAN) in general and NAPLAN Writing Assessment in particular are explained and an example of NAPLAN Writing assessment prompt is given as well. The research question is divided into a number of research issues so as to make it more achievable. A research objective is then specified for each research issue. The various steps involved in each of the research objectives are detailed. Finally, the research methodology adopted to develop the solution for the research question is outlined.

In the next section, the definition of writing genre and its various types are explained.

## 3.2. Writing Genre and its various types

Writers write for a variety of reasons, some of which are to tell a story, to convey meaning or their own personal opinion about a topic or to explicate a certain subject or idea. The purpose for which a piece of text is written essentially determines the writing genre of the text. Formally, [1] defines writing genre as the style in which a writer chooses to present textual content to the reader. Broadly speaking, essay writing has four types of writing genres. The four most common writing genres are:

1. Narrative writing. In narrative writing, the writer tells a story or part of a story, mainly from the viewpoint of central character/characters, for example, NAPLAN narrative writing.
2. Persuasive writing. In persuasive writing (also called argumentative writing), the purpose of the writer is to convince the reader of his opinion or viewpoint, for example, GMAT AWA.
3. Descriptive writing. In this genre style, the writer tries to describe a location/person/situation in such great detail that the reader can 'visualise' it for himself, for example, writing which depicts oceanic or mountainous views.
4. Expository writing. This style is used mainly for explaining a certain idea or topic, for example, technical articles and help manuals.

Since writing genres are not mutually exclusive of each other, an essay can have content that fits into more than one genre. However, if the question prompt is designed specific to a genre, then the essays written should be in accordance to the genre in question.

For the purpose of this thesis, narrative writing is of most interest. In the next section, the reason and motivation for this thesis is outlined.

## 3.3. Motivation for this thesis

Generally, the essay scoring process is time-consuming and labour-intensive. It is the norm for high-stakes assessment to score each essay at least twice. According to estimates published in [2], the cost of scoring a holistic rubric-based essay at the rate of 12 minutes would involve cost depending on hours taken for the essays. This cost would increase further if other factors such as essay length, number of raters required and administration of feedback are taken into account. Further, for essays based on a multi-trait scoring rubric, the cost would increase manifold because of the factors associated with such scoring, as discussed in the previous chapter.

In the context of National Assessment Program – Literacy and Numeracy (NAPLAN) marking in Australia, a trained marker is expected to complete around 10 essays per hour. This is challenging and requires a considerable effort from the marker. Moreover, the NAPLAN writing assessment test is administered to students in May every year and the results are available no earlier than August. The time-lag between the administration of the test and the reporting of the results can be attributed to the mammoth effort involved in a national-level assessment. If an AEG system is used to score the NAPLAN essays, there is a fair possibility that the human workload can be decreased significantly and the results can be reported faster. The aforesaid reasons are the motivation for this thesis.

In the next section, problem overview of this thesis is discussed and the problem that this thesis hopes to address is formally defined.

## 3.4. Problem Overview and Problem Definition

In this section, an overview of the research problem is discussed. From an extensive review of the existing literature of AEG systems, we have identified that there is a need for an AEG system which:

- can grade essays in the English language;
- can perform analytic scoring to assign an individual score to each criterion of the essay;
- can assess and score narrative essays;
- can model both the linear as well as the non-linear nature of relationships between the 10 features of the essay as outlined by NAPLAN and the essay grade;
- can be trained and calibrated using a dataset of less than 200 essays to make the system more feasible and usable;
- is computationally non-intensive so as to promote general and real-time use; and
- can handle improperly constructed responses

As discussed in chapter 2, some of the existing AEG systems have been developed for grading essays in the English language while other systems were developed for grading essays in other languages. However, for the purpose of this thesis, we are interested in an AEG system

for grading essays in the English language. Further, in the existing systems, AEG systems for performing analytic scoring are very few when compared to AEG systems for performing holistic scoring. Analytic scoring means that a score is assigned separately to each criteria of the essay such as spelling, vocabulary and sentence structure; whereas holistic scoring means that a single overall score is assigned to the essay. As discussed in chapter 2, only four of the existing AEG systems – Intellimetric, My! Access, Jess and CarmelTC, can be used for multi-trait scoring. All the other systems can be used mainly for holistic scoring. Of the AEG systems which are capable of performing analytic or multi-trait scoring, all are not capable of scoring narrative essays. This is because scoring narrative essays is a very complex and challenging task. Furthermore, most of the existing AEG systems assume that the essay features and essay grade have a linear relationship, which is not necessarily true. In some cases, the relationship might be non-linear. For example, an essay that has a high number of nouns does not necessarily display a higher vocabulary. Hence, awarding a high vocabulary score to that essay will not be fair. On the other hand, an essay that has a higher number of adjectives can be awarded a higher vocabulary score. Hence, an AEG system needs to model both the linear and the non-linear nature of relationships between the essay features and the grade. In cases where the relationship between the feature vector and the essay grade might be non-linear, the existing AEG systems do not provide a methodology to model the same. As mentioned in chapter 2, the AEG systems that model both the linear as well as the non-linear nature of relationships between the essay features and its grade are Intellimetric, My! Access, IEMS, PS-ME and the AEG system that uses unsupervised learning, based on a voting algorithm. Of these, only two AEG systems, Intellimetric and My! Access, can grade narrative essays in an analytic scoring system.

For training and calibration of the AEG system, a relatively small dataset of less than 200 essays is desired. This is because it is unrealistic to train the AEG system using a few hundreds

or thousands of essays when only a few hundred essays need to be tested. However, neither of the two AEG systems which have so far satisfied the other criteria, satisfy this criterion. Yet most of the other AEG systems satisfy this criterion. Additionally, it is desirable that the AEG system is computationally non-intensive. This is to ensure that the AEG system can be used in real-time. However, only about 7 existing AEG systems are computationally non-intensive. Finally, most of the existing systems assume that the AEG system will receive only properly constructed responses and hence they do not propose a methodology to handle improperly constructed responses. But in reality, there can be a variety of improperly constructed responses in the essay dataset, which need to be handled properly by the AEG system.

According to [3], in order to be feasible, the AEG system should be "able to preserve the benefits of student responses; should increase essay scoring through-put; should reduce grading costs; and should accurately assess and grade essays".

In light of the above discussion, the problem definition for this thesis can be formally stated as:

*"How can a robust methodology for automated essay grading be developed to grade narrative essays and assign analytic scores?"*

In order to answer the research problem, it is beneficial to first study the nature of narrative writing in NAPLAN. Therefore, in the next section, an overview of the NAPLAN writing assessment is provided along with an example of NAPLAN question prompt for narrative essay writing.

## 3.5. Overview of NAPLAN Writing Assessment

The National Assessment Program – Literacy and Numeracy (NAPLAN) is an annual written English assessment for school students in Years 3, 5, 7 and 9 in Australia [4]. The assessment is held for all students simultaneously, every year in May. The four main areas that are assessed are reading, writing, language conventions (spelling, grammar and punctuation) and numeracy. In order to prepare students for the NAPLAN, several practice tests are given as mock tests in class prior to the actual test.

In the NAPLAN writing assessment, a question prompt is given to the students and the nature of the response required is explained to them. After the completion of the test, the responses are collected from all the schools and sent to marking centres for marking. Trained markers assess the responses using the NAPLAN Markers Guide which is a NAPLAN writing assessment guide [5].

### 3.5.1. Question Prompt in NAPLAN

A NAPLAN question prompt can be described as a "narrow and constrained task", as per the definition in [6]. A sample prompt is shown in figure 3.1. The prompt consists of a collage of several photographs depicting the same theme/idea. In controlled examination environments in class rooms, with teachers as invigilators, students are asked to construct and write narrative responses based on the theme in the question prompt. This annual writing assessment is held consecutively for year levels 3, 5, 7 and 9 and the same prompt is used for everyone. The advantage of using the same prompt is that writers of different levels of proficiency can use their creativity and demonstrate their writing skills. Students of different year levels use different words, a variety of sentence structures and establish their level of grammar while writ-

ing short stories in response to the same prompt [7]. The advantage is that the constructed responses for the same prompt can be evaluated using the same rubric nation-wide and thereafter, the evaluation results can be used for a variety of diagnoses ranging from the analysis of the performance of the student to the relative performance of the student with other students in the same year level within the same school and in other schools. Furthermore, using the evaluation results, a variety of analyses can be performed, relating to the performance of a school within a state to the performance of schools in similar socio-economic settings across Australia.


**Figure 3.3: A sample NAPLAN question prompt**

### 3.5.2. Writing Assessment Criteria in NAPLAN

The marking of the students' responses in the NAPLAN writing assessment is carried out by human markers and is based on the NAPLAN Narrative Marking Guide **[5]**. NAPLAN writing assessment is a multi-trait scoring method which means the essay is scored on various

traits inherent in the text and each trait is scored according to a pre-defined scale. It aims to evaluate an essay individually in ten criteria as listed below from the NAPLAN Narrative Marking Guide.

Audience – the writer's capacity to orient, engage and affect the reader

Text Structure – organization of narrative features in an appropriate and effective structure

Ideas – creation, selection and crafting of ideas

Character & Setting – portrayal of character and/or development of a sense of place, time and atmosphere

Vocabulary – the range and precision of language choices

Cohesion – the control of multiple threads and relationships

Paragraphing – segmenting of text into paragraphs that assist in reading

Sentence Structure – production of grammatically correct, structurally sound and meaningful sentences

Punctuation – use of correct and appropriate punctuation

Spelling – accuracy of spelling and difficulty of words used

Each criterion has a score band associated with it. The score band ranges between 0-6, as shown in table 3.1.

**Table 3.11: NAPLAN Writing Assessment rubric elements and their score ranges**

| Audi-ence | Text Struc- | Ide as | Char-acter | Vo-cab | Cohe-sion | Para-graphing | Sen-tence | Punctua-tion | Spelli ng |
|---|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | ture | | and setting | | | | struc-ture | | |
| 0-6 | 0-4 | 0-5 | 0-4 | 0-5 | 0-4 | 0-2 | 0-6 | 0-5 | 0-6 |

When the essay is assessed for the criteria 'Audience' using the marking guide then the assigned score would be in the range of 0-6 inclusive of both numbers. Similarly, the score assigned for the criteria 'text structure' would be in the range of 0-4 inclusive of both numbers. In this way, scores are assigned to each of the ten criteria using the pre-defined score ranges. Ultimately, the final essay grade is simply the sum of the scores of each criterion. We divide the ten criteria into four broad categories, as illustrated in figure 3.2.

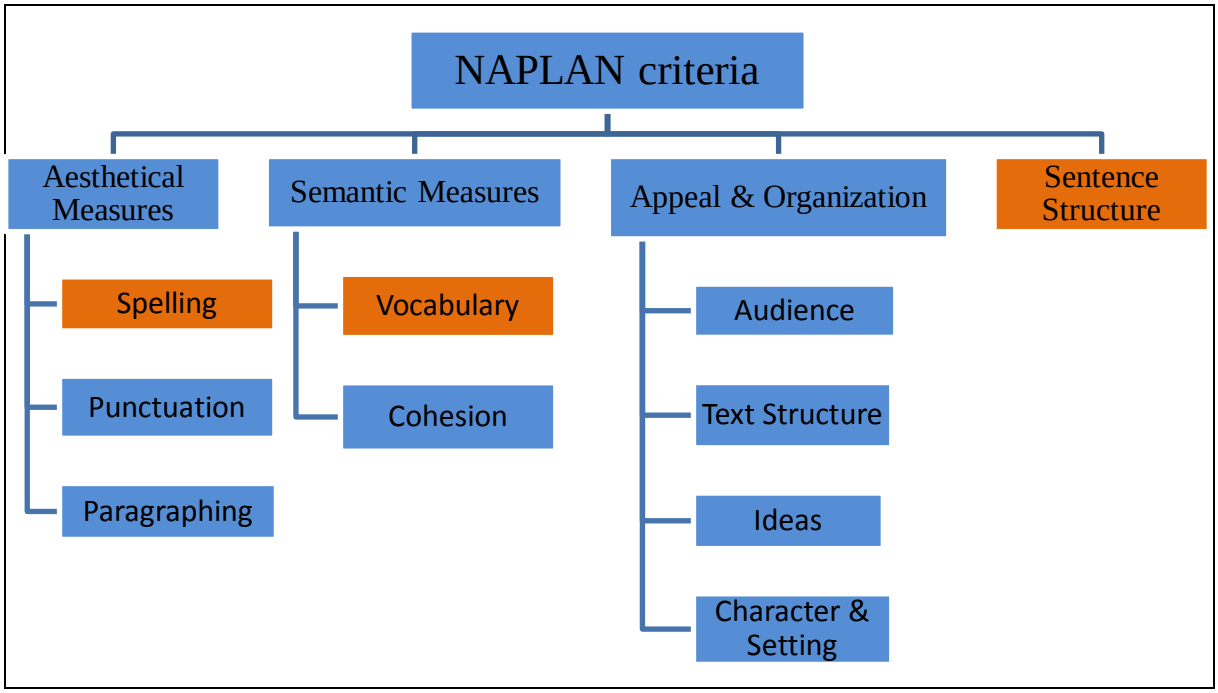



**Figure 3.4: Categories in NAPLAN Writing Assessment Criteria**

The four broad categories are aesthetical measures, semantic measures, appeal and organization and finally, sentence structure. They are defined as follows:

Aesthetical measures: The surface features of an essay that promote ease of reading are called the aesthetical measures. In the context of NAPLAN, spelling, punctuation and paragraphing come under this category.

Semantic measures: The features of the text that contain the content words and an examination of the relationships between the content words are called semantic measures. In the context of NAPLAN, vocabulary and cohesion come under this category.

Appeal and Organization measures: The features of the text that illustrate the overall appeal, ideas and their proper flow are called appeal and organization measures. In the context of NAPLAN, audience, text structure, ideas, character and setting come under this category.

Sentence Structure: The feature of the text targeted at the production of correct and meaningful sentences is sentence structure. In the context of NAPLAN, this feature is a broad category in itself.

For the purpose of this thesis, we restrict our scope to grading the three criteria of spelling, vocabulary and sentence structure. Having analysed the nature of NAPLAN question prompt and its writing assessment criteria, it is now time to coin an answer to the research problem mentioned in section 3.4. As an effort in this direction, the specific research issues are listed in the next section.

## 3.6. Research Issues

Based on a thorough review of the existing body of literature on automated essay grading, the research question has been identified, as previously stated in section 3.4. In order to address the problem, the following research issues have been formulated:

1. To develop an AEG system which is capable of grading essays in the English language and can handle improperly constructed responses.
2. To develop modules for the analytic scoring of narrative essays. The modules should be able to model both the linear and non-linear relationships between the essay features and its grade, should be computationally non-intensive and should be able to be trained and calibrated using a relatively small dataset. This research issue is further sub-divided into three research issues as follows:
    i. Develop a module for grading the spelling criterion, according to the NAPLAN rubric.
    ii. Develop a module for grading the vocabulary criterion, according to the guidelines stated by the NAPLAN rubric.
    iii. Develop a module for grading the sentence structure criterion, according to the NAPLAN rubric.
3. To verify and validate the methodologies developed in each of the research issues above.

Each of these research issues is explained in detail with the help of research objectives in the next section below.

## 3.7. Research Objectives

In light of the research issues outlined above, this thesis aims to achieve the following objectives:

### 3.7.1. To develop an AEG system that is capable of handling improper responses.

The subtle assumption in many AEG systems, such as PEG and LSA, that an unscored essay reflects a properly constructed response [3], is not realistic. This is because students in primary school are still learning the basics of language and the specifics of writing a constructed response such as an essay, hence there are instances when they end up writing improperly constructed responses. These include drawing a picture instead of writing an essay, writing gibberish words and/or sentences and writing text containing so many spelling and grammar errors that it is indecipherable and incomprehensible. Such essays are attributed to the 'noise' in the dataset.

Futagi highlights the importance of the detection of noise in essays, which is an often-ignored topic when discussing the automated scoring of essays [8]. The attributes of misspellings, context-based spelling errors, morphological- or syntax-based errors and punctuation errors in the essays to the 'noise'. He claims that noise in essays has been shown to have a detrimental effect on the design and implementation of an automated collocation detection tool for use by Educational Testing Service (ETS). His work stresses the fact that if an automated essay scoring system is built without initially filtering the noise from the essays, then it has a detrimental effect on the performance of the system. This conclusion is also supported by [8]. Hence, an AEG system is required that has no prerequisite of properly constructed responses.

It is acknowledged by many researchers that essays contain noise that makes them unsuitable for grading unless they are filtered out manually, which in turn involves huge costs [9]. However, there are very few details on how to detect noisy essays and cleanse them. The challenge here is to identify such essays in the dataset and highlight them so they can be given to a human marker for scoring. Further, using the available resources instead of developing new technology, these essays should be separately graded from the dataset that is graded by the system so as to reduce the impact of the noisy essays on the performance of the AEG system.

### 3.7.2. To develop modules for analytic scoring of narrative essays.

Narrative essays are viewed as being the most difficult to grade automatically, as mentioned previously. This is because narrative essays are typically creative writing or stories built around a central theme but told in the author's own style.

The existing AEG systems need relatively large datasets for training purposes, typically in the range of 270 to a few thousand (for example, BETSY). This requirement is a huge deterrent and is unpractical if only a few hundred essays are to be graded. Hence, it is desired that the AEG system should be able to grade essays by using a relatively small dataset, preferably less than 200 essays.

Thorough studies of the existing AEG systems lead us to conclude that there is no generic model for analytic scoring. As discussed above, a generic model is not prompt-specific and hence will not have to be re-trained for every new prompt. Instead, it will be trained only once using a dataset from a certain genre and then the system can be used for grading essays for any prompts within that genre. This helps in reducing computational time, resources and ultimately, cost.

Many available AEG systems are trained on well-structured essays, such as newspaper articles or academic materials such as text books. These materials consist of correct spelling,

proper usage of grammar and sentence structures. Some AEG systems expect the students to check their own spelling before submitting their essays while many other systems allow students to use spell checking features available in word processing programs. All these practices are not useful when we are trying to grade students' written essay responses from primary school through to high school.

There is no existing automated essay grading system that can be used to grade essays according to NAPLAN rubric.

The NAPLAN rubric consists of clearly stated rules and directions on how to score an essay in various criteria. The rubric is a guideline for human markers and so far, there has been no attempt to computerize the rubric completely. A partial attempt towards assessing the content of the essays was done in the MarkIT project, as pointed out in chapter 2, section 2.6.2.4. Although MarkIT tried to evaluate the essay content and present it visually, it expects the human markers to assign scores to the various criteria. Hence, an automated system that is capable of assigning scores to the criteria of spelling, vocabulary and sentence structure, according to the NAPLAN rubric, is required.

Moreover, the problem in grading spelling according to the NAPLAN rubric is that there is usually disagreement between markers regarding the class of a word. If one marker thinks that a particular word should be classified as common, another marker in another corner of Australia may think that the same word should be classified as difficult. WADET reported last year that the inconsistency in classifying words led to the incorrect assignment of scores while marking spelling. Since NAPLAN is a national assessment, it is imperative that the same rubric and the same marking criteria are applied all over Australia.

Further, while marking spelling, the number of words in each class is considered in order to assign the final mark for the essay. To do this, human markers are expected to count words

manually from the computer screen. But due to the difficulty in doing so, markers do so only sometimes, thus there is a possibility that the scores might be biased towards a marker's feeling of *how many words are present in each class.*

We aim to design a model that is capable of scoring narrative essays. In order to make automated essay grading a more practical and feasible solution, it is imperative that the new model aims to overcome the research gaps stated above. Hence, the new model should require a relatively small dataset for training. Further, it should try to produce the best results by making use of minimum resources. By reducing the computational load, we aim to reduce the costs involved. We aim to develop a methodology that can capture both the linear as well as the non-linear relationships between the feature vector and the essay grade. We aim to formalize the NAPLAN criteria of spelling, vocabulary and sentence structure for the purpose of building the above model. Using these modules, narrative essays can be scored automatically.

#### 3.7.2.1. To develop a module for grading spelling

According to the NAPLAN rubric, to grade spelling, the skill focus is on the accuracy of spelling and the difficulty of words that are used by the student in the essay. In the context of NAPLAN, words are divided into four classes depending on various factors including the difficulty of the spelling. The four classes are simple, common, difficult and challenging. Some examples for each class are provided in table 3.2.

**Table 3.12: Word class and examples according to the NAPLAN classification**

| Class | Examples |
|---|---|
| Simple | I, a, am, me, but, bad, drop, glass, school |
| Common | Air, any, catch, middle, hospital, happening |

| | |
|---|---|
| Difficult | Obese, chocolate, generate, invisible, community |
| Challenging | Baulk, brevity, guarantee, responsibility, leisure |

NAPLAN provides a database of about 1200 classified words but it is by no way comprehensive. Human markers use the list as a reference and classify new words based on:

1. the classification of a word that is similar to the new word; and
2. own interpretation and conception regarding the class of the word.

Since the interpretation of one human marker varies from the other, the scores vary from one human marker to the other. NAPLAN has reported that this issue has led to the incorrect scoring of spelling and non-uniform scoring across Australia. For example, if a human marker classifies 'mineral' as common instead of difficult, the score would be one point less. On the other hand, another human marker who correctly classifies 'mineral' as difficult would be able to assign the correct score.

The following table 3.3 describes further, the skill that is to be demonstrated by the student in one's spelling in order to gain a particular score [5].

**Table 3.13: Score Descriptors for 'Spelling'**

| Score | Description |
|---|---|
| 0 | No conventional spelling |
| 1 | Few examples of conventional spelling |
| 2 | Correct spelling of most simple words and some common words (errors evident in common words) |

| | |
|---|---|
| 3 | Correct spelling of most simple words and most common words |
| 4 | Correct spelling of all simple words, most common words and some difficult words (errors do not outnumber correct spellings) |
| 5 | Correct spelling of all simple words, most common words and at least 10 difficult words (errors do not outnumber correct spellings) |
| 6 | Correct spelling of all simple words, all common words, at least 10 difficult words and some challenging words (occasional minor errors-typos are disregarded when assigning this category) |

As shown in table 3.3, an essay would be assigned a score of '0' if there is no conventional spelling in it or if there are no proper words in it. For an essay to be assigned a score of '1', there has to be few examples of proper words in conventional spelling. To be assigned a score category of more than '1', the class of each word in the essay is to be determined. Depending on the correctness of the words and the classes of the words, a score category of '2' or '3' can be assigned. However, for the score of '2', the essay should have some incorrectly spelt words belonging to the class 'common'. For the score of '3', most (atleast 80%) simple words and most common words should be spelt correctly. When there are correctly spelt words in the essay belonging to the class 'difficult', it can be assigned a score category of atleast '4' depending on the other conditions being met. For a score category '4', the essay should have correct spelling of all simple words, most common words and some (2-3) difficult words; with the condition that the errors in difficult words do not outnumber the correctly spelt difficult words. To be assigned a score category '5', the essay should have correct spelling of all simple words, most common words and at least 10 difficult words; with the condition that the errors in difficult words do not outnumber the correctly spelt difficult words. Finally, to be assigned the highest score of '6' in this criterion, the essay should have correct spellings of all

simple words, all common words, at least 10 difficult words and some (2-3) challenging words. When assigning this score category, occasional errors such as typos are ignored.

When developing the module to grade spelling automatically, the aim is to develop a methodology that can first classify a word correctly into one of the four classes and then assign a score based on the above guidelines. The difficulty here is how to automatically classify a word that has been incorrectly spelt. Although a human marker can interpret the correct word that the student intended to write, the computer needs to perform some analysis in order to do the same. We aim to do this by making use of available resources and by avoiding heavy computations.

#### 3.7.2.2. To develop a module for grading vocabulary

According to the NAPLAN guide, to grade vocabulary, the skill focus is on the range and precision of language choices. The range refers to the various parts of speech evident in the essay. The precision refers to the appropriateness and effectiveness of the word in how well it matches the writing genre. The different parts of speech in the English language and their examples are listed below.

1. Noun – a word that names a person, place, thing or a concept, for example, James, New York, chair, hope.
2. Pronoun – a word that replaces a noun or noun group, for example, she, these, who, which
3. Adjective – a word that gives additional information about the noun, for example, 'It is a stubborn stain', where 'stubborn' is the adjective.
4. Verb – a word that describes an action or gives a sense of what is happening, for example, running, swam, laughed.
5. Adverb – a word that provides additional information about the verb or adjective, for example, walked slowly, ran away.

6. Preposition – a word that denotes position, for example, below, under, over and around.
7. Conjunction – a word or words that join(s) two or more words, for example and, not only…but also.
8. Interjection – a word that expresses sudden emotion, for example, Alas! , Oh!

Vocabulary can be assigned a score between the range of 0-5 inclusive of both numbers. In table 3.4, the score descriptors are given for each score [5].

**Table 3.14: Score descriptors for 'Vocabulary'**

| Score | Description |
|---|---|
| 0 | Symbols or drawings |
| 1 | Very short script |
| 2 | Mostly simple verbs, adverbs, adjectives or nouns<br><br>May include two or three precise words |
| 3 | Precise words or word groups (may be verbs, adverbs, adjectives or nouns) |
| 4 | Sustained and consistent use of precise words and phrases that enhance the meaning or mood |
| 5 | A range of precise and effective words and phrases used in a natural and articulate manner<br><br>Language choice is well matched to genre. |

As shown in table 3.4, to be assigned a score of '0', the essay should have either symbols or drawings but no examples of proper words. If the essay is a very short script of about 2-3 sen-

tences with few examples of conventional words then it can be assigned a score of '1'. If there are more than 2-3 sentences in the essay, then the essay can be assigned a score of more than '1' depending on the other conditions being met. In such cases, the part of speech of each word is determined. If the essay has mostly simple nouns, verbs, adjectives or adverbs and two or three precise words then it can be assigned a score of '2'. If the essay has more than three precise words and words which are mostly nouns, verbs, adjectives or adverbs, then it can be assigned a score of '3'. If the essay contains phrases intended to enhance the meaning of the text and such usage is sustained and consistent then it can be assigned a score of '4'. To be assigned the highest score of '5', the essay should have a range of effective words used in a natural and articulate way such that the language is well matched to the genre.

To develop the module for grading vocabulary automatically, the aim is to develop a methodology that can capture the above guidelines effectively and assign a vocabulary score accurately. In order to do this, we need to identify the parts of speech of words used in this essay and then analyze the precision of the words. We expect to encounter difficulty in analyzing the precision of the words because of inherent spelling errors and also because students in primary school are still learning and experimenting with language and are bound to make mistakes when they try to spell a new word. For example, a primary school student could spell 'malicious' as 'malishus'. Hence, our system will have to identify the quantity and quality of words used, despite the spelling errors.

#### 3.7.2.3. To develop a module for grading sentence structure

According to the NAPLAN guide, to grade sentence structure, the skill focus is on the production of grammatically correct, structurally sound and meaningful sentences. In the English language, a sentence is defined as a group of words that make complete sense. It can be a

statement (It is your box), a question (Is it your box?), a command (Take your box!) or an exclamation (What a nice box!). There are three types of sentences as follows:

1. Simple sentence - consists of a single clause, for example, '*We took the box*'.
2. Compound sentence - consists of two or more clauses which are coordinated or linked so that each clause has equal status, for example, '*We took the box and painted it*'.
3. Complex sentence - contains embedded and/or subordinate clauses. Embedded clauses are those that are as a part of another clause and so have no relation with the main clause, for example, '*We took the box and painted it with oil paints*'.

Apart from the correctness of the different types of sentences, the variety in sentence formations is also considered. Simple, compound and complex sentences can each have either basic or sophisticated structures. Sophisticated structures use more phrases than basic structures. Sentence structure can be assigned a score of 0-6 inclusive of both numbers. In table 3.5, the score descriptors for each score category are given [5].

**Table 3.15: Score descriptors for 'Sentence Structure'**

| Score | Description |
|---|---|
| 0 | No evidence of sentences |
| 1 | Some correct formation of sentences<br><br>Some meaning can be construed |
| 2 | Most simple sentences are correct<br><br>Meaning is predominantly clear |
| 3 | Most simple and compound sentences correct |

| | |
|---|---|
| | Some complex sentences are correct<br><br>Meaning is predominantly clear |
| 4 | Simple and compound sentences are correct<br><br>Most complex sentences are correct (OR)<br><br>All sentences correct but do not demonstrate variety<br><br>Meaning is clear |
| 5 | Sentences correct (allow for occasional type, ex. a missing word)<br><br>Demonstrates variety in length, structure and beginnings<br><br>Meaning is clear and sentences enhance meaning |
| 6 | All sentences are correct<br><br>Writing contains controlled and well-developed sentences that express precise meaning and are consistently effective. |

As shown in table 3.5, to be assigned a score '0', the sentence would not have any sentences at all. If there are some sentences in the essay and some meaning can be interpreted then the essay can be assigned a score '1'. If most (80%) of the simple sentences have correct formations and their meaning can be understood then the score '2'can be assigned. When there are all three types of sentences in the essay, then it can be assigned a score of '3' or more depending on the other conditions being met. For a score '3', most simple and compound sentences are correct, some (2-3) complex sentences are correct and the meaning is mostly understandable. For a score '4', one of the two conditions need to be met: (1) all the simple and compound sentences are correct and most (80%) complex sentences are correct or (2) all the

sentences are correct but they do not demonstrate variety. The meaning of all the sentences has to be clear. When there is a variety in length of sentences, the structure of sentences and the sentence beginnings then the essay can be assigned a score of '5' or '6' depending on the other conditions being met. For a score '5', the essay should have all sentence formations correct and the sentences should be meaningful and clearly understandable. When assigning this category, an occasional typo error such as a missing word is ignored. To be assigned the highest score '6', all the sentences in the essay need to be correct. Additionally, they have to be well developed, precise, meaningful, effective and clearly understandable.

To develop the module for grading sentence structure automatically, the aim is to develop a methodology that can capture the above guidelines and assign scores with accuracy. The module should be able to detect if there are any sentences in the essay. If there are, then for each sentence, the type of sentence and the correctness of the sentence need to be determined. Depending on the number of correct sentences in each type and provided that the number of correct sentences in that type is more than the number of incorrect sentences, the score for sentence structure is to be assigned. The findings of [6] are in support of our AEG system because the NAPLAN writing test is specific and constrained in nature, similar to the essay questions in GRE and GMAT AWA exams. So the agreement between our AEG system and the human marker would be high.

### 3.7.3. Validation of the proposed methodologies

In this step, the verification and validation of the proposed methodologies will be undertaken.

To detect improperly constructed responses, a rule-based approach will be developed. The coding of the algorithm will be in the Java language and then the simulations will be performed using a real world essay dataset provided to us by WA-DET.

For the purpose of grading spelling, heuristics and rule-based approaches will be developed and coded using Java language. The validation and verification will be carried out using the NAPLAN database and real world essays as the test dataset.

For the purpose of developing the vocabulary module, heuristics and rule-based approaches will be developed and coded in Java language. The validation and verification will be carried out using a real world essay dataset provided to us by WA-DET. Furthermore, another approach to model the linear and non-linear relationship will be developed using neural networks and simulations will be conducted using MATLAB software on a real world essay dataset.

The sentence structure module will initially be developed using a rule-based approach using Java software. At a later stage, a neural network model will be created and simulations will be carried out using MATLAB software. Finally, the verification and validation of the methodology for grading sentence structures will be undertaken for both approaches using a real world essay dataset.

In the next section, the choice of research methodology to achieve the research objectives is discussed.

## 3.8. Choice of Research Methodology

To achieve the aforesaid objectives, a system development-based research paradigm will be utilized that aims to create innovations by first developing ideas to solve a problem, building

a conceptual framework, developing a system architecture and building the system and then evaluating it [11]. Specific to my research, the above-mentioned stages are categorized into three broad levels as discussed below [11]:

1. conceptual level: creating new ideas and new concepts after the literature review and a thorough analysis of the existing systems. This has been carried out in Chapter 2.

2. perceptual level: formulating a new scoring method through designing and building the tools, environment or system through implementation.

3. practical level: carrying out testing and validation through experimentation.

The research methodology employed in this thesis is in accordance with the abovementioned levels and will be implemented in four phases as follows:

**Phase 1 - Literature review, Identification of Issues and Problem Formulation**

As an outcome of this phase, the research issues were identified and the research objectives were formulated in Chapter 2.

**Phase 2 – Development of methodology for linear and non-linear mapping of input feature elements to output essay grade**

This phase too falls under the conceptual level. It is sub-divided into two steps as follows:

**Step 2.1-Design of Feature Vector of the rubric element**

In this step, the feature vector of the rubric element is designed such that it takes into account all the specifics of the elements as mentioned above. For example, for the rubric element 'vocabulary', the feature vector would contain some of the specifics mentioned in table 3.4 such as the number of nouns, verbs, adverbs, adjectives and prepositions in the essay. Similarly, for the rubric element 'sentence structure', the feature vector would contain some of the spe-

cifics mentioned in table 3.5 such as the number of simple sentences, number of compound sentences, number of complex sentences and number and type of incorrect sentences.

**Step 2.2-Mapping of Input Feature Vector to an Output Essay grade**

The mapping process is illustrated in figure 3.3, showing the design of a multi-layer feed forward neural network [12] with a back propagation algorithm [13]. This neural network has three layers : the *Input layer* consists of input nodes, the *Hidden layer* consists of nodes that are responsible for the mapping process and the *Output layer* provides the output of the mapping process. Each connection between nodes has an associated weight, denoted by 'W' and 'w'.

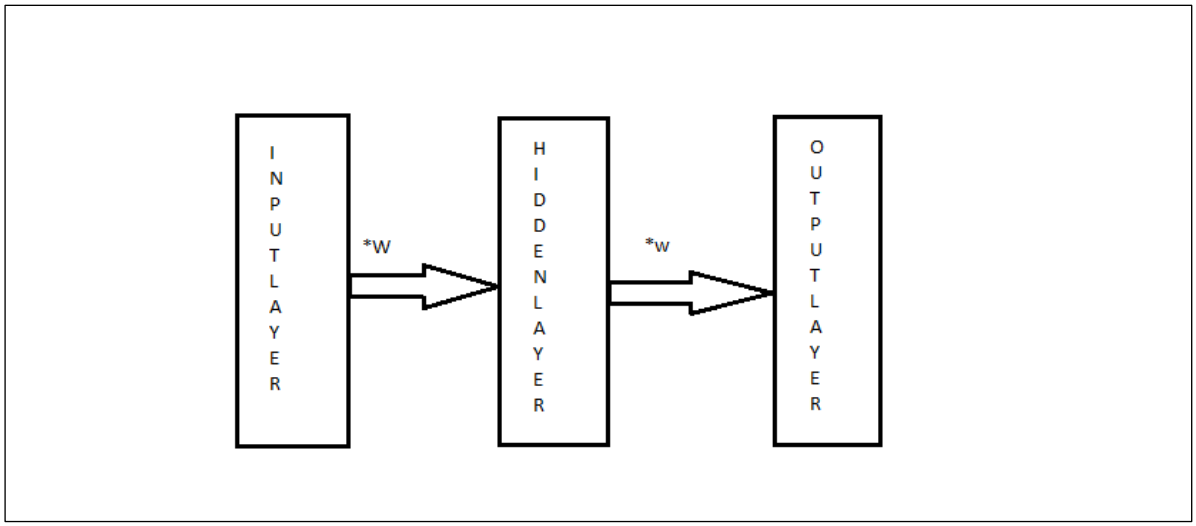



**Figure 3.5: Multi-layer Feed Forward Neural network architecture [10]**

Numerical values corresponding to the feature vector specifics of the rubric element are fed into the input layer. Processing takes place in the hidden layer (explained in detail below) and the output, that is the essay grade, is fed into the output node. There are three phases in the mapping process: the analysis phase, the training phase and the grading phase. A large set of essays (pre-scored by human experts) and their grades have been supplied to us by WADET. This set is called the training set and is used in the analysis phase and training phase.

In the analysis phase, for every essay from the training set, the feature vector (described above in Phase 2.1) is derived. This means that values are derived for each specific of the rubric element. In the training phase, the value of the feature vector of an element (derived in the analysis phase) is fed into the input layer of the neural network, the network processes it and it is iteratively trained until it performs optimally. Finally, in the grading phase, a new test essay is graded. The feature vector of the new essay is fed into the network and the suitable grade is assigned by the system.

**Phase 3 – Detailed Methodology Development**

During this phase, the mathematical model that underpins the methodology is developed in detail. This phase falls under the perceptual level of the chosen research approach.

**Phase 4 – Verification and Validation of the proposed methodology through testing**

In this phase, the trained neural network model is used to determine the essay grade and to compare it with the human grade. This corresponds with the practical level of the research methodology.

In the next section, a summary of the main points discussed in this chapter is mentioned and the chapter is concluded.

## 3.9. Conclusion

In this chapter, the research question that this thesis will endeavour to address was identified. The research question was broken down into a number of research issues to make it easier to develop the solution. The research objectives for each of the research issues were presented after which, the research methodology that will be adopted to achieve the research objectives was explained. Furthermore, the research methodology adopted for this research was described in detail and the various stages and the actions performed in each stage of the research methodology were elucidated.

In the next chapter, the detailed conceptual level framework of the proposed AEG system will be provided as well as the working of the various modules that comprise the AEG system.

## 3.10. References

[1] H. W. Lam, T. Dillon, and E. Chang, "Determining Writing Genre: Towards a Rubric-based Approach to Automated Essay Grading," presented at the IEEE International Conference on Advanced Information Networking and Applications (AINA), Singapore, 2011.

[2] R. A. Hardy, "Examining the cost of performance assessment," *Applied Measurement in Education,* vol. 8, pp. 121-134, 1995.

[3] K. W. K. Chung and H. F. O'Neil, "Methodological approaches to online scoring of essays," University of California, Los Angeles, Technical Report no. 461, 1997.

[4] Australian Government. (2011, 13 October 2011). *National Assessment Program*. Available: http://www.nap.edu.au/NAPLAN/index.html

[5] Department of Education, *Narrative Marking Guide 2010, National Assessment Program-Literacy and Numeracy*: Government of Western Australia, 2010.

[6] D. McCurry, "Can machine scoring deal with broad and open writing tests as well as human readers?," *Assessing Writing,* vol. 15, pp. 118-129, 2010.

[7] E. Cotos and N. Pendar, "Automated diagnostic writing tests: Why? How?," in *Towards Adaptive CALL: Natural Language Processing for Diagnostic Language Assessment*, Ames, Iowa, 2008.

[8] Y. Futagi, "The effects of learner errors on the development of a collocation detection tool," presented at the Fourth workshop on Analytics for noisy unstructured text data, Toronto, ON, Canada, 2010.

[9] O. Mason and I.-G. Stephenson, "Automated free text marking with Paperless School," presented at the Sixth International Computer Assisted Assessment Conference, Loughborough, UK, 2002.

[10] Madhavi Ganapathiraju, N. Balakrishnan, Raj Reddy and J. Klein-Seetharaman, "Transmembrane helix prediction using amino acid property features and latent semantic analysis," *BMC Bioinformatics,* vol. 9, 2008.

[11] R. D. Galliers, *Information Systems Research: Issues, Methods and Practical Guidelines*: Blackwell Scientific Publications, 1992.

[12] T. S. Dillon and Niebur, D., Ed., *Ne*ural networks applications in power systems, CRL Publishing Ltd, Market Harborough, UK, 1996.

[13] S. Haykin, Neural networks: A comprehensive foundation: Prentice Hall, 1999.

# Chapter 4: Overview of Conceptual Framework

## 4.1. Introduction

In the previous chapter, the research gaps that were identified as a result of the literature survey were discussed. Then, the research aims of this thesis were presented, after which the objectives that we aim to achieve were recapped. The development of a generic model for scoring narrative essays is the aim of this thesis. The features of this model include the following:

- genre-specific but not prompt-specific
- able to achieve accurate results with minimum training to keep the set-up costs to a minimum
- need a relatively small dataset for training to keep the set-up costs to a minimum
- use available resources rather than developing new resources
- able to handle improperly constructed responses in addition to properly constructed responses
- able to capture both linear and non-linear relationships between the essay features and grades, and finally
- should not be computationally intensive so as to promote general use

The other aims of this research are to develop a module to grade spelling, vocabulary and sentence structure, according to the NAPLAN rubric.

In this chapter, an overview of the conceptual framework of the proposed AEG system is presented. The conceptual framework is composed of two important and sequential processes: the pre-processing stage, called the filter process, and the essay grading process. An overview of both processes and a detailed explanation is provided in this chapter with a description of the essay grading process itself which comprises three separate modules, one each for grading spelling, vocabulary and sentence structure. Also in this chapter, the methodology and working of each module is explained in detail.

In the next section, an overview of the proposed AEG system is presented.

## 4.2. Overview of the proposed AEG system

To build a successful AEG system, it is essential that the sample data set is studied. Through close, manual observation of the essay dataset, heuristics are developed. Heuristics are solid assumptions which are based mostly on common sense. There have been a few instances of the use of heuristics and rule-based modules in an AEG system. [1, 2] use a large number of heuristics to develop the Argument Partitioning and Annotation (APA) module which can be used in an AEG system. The APA module uses dictionary words, terms and other lexical cues and searches for them in the text. Then it analyses the syntactic structures of sentences and finally annotates and highlights the arguments in the text. The results obtained in this research highlight that using heuristics can actually improve the working of AEG systems, as will be discussed in the subsequent chapters.

Figure 4.1 illustrates the conceptual framework of the proposed AEG system. The essay dataset consists of handwritten student essays from Years 3, 5, 7 and 9 from the Department of Education, Western Australia. Generally, the input to an AEG system consists of essays that have been input by using the standard keyboard. For handwritten essays, the approach is very challenging because it requires special Optical Character Recognition (OCR) systems to convert them into a computer-readable format. This technology is still inaccessible to school children in most countries and moreover, is very expensive. The students' handwritten essays are typed and produced as documents in Microsoft Word 2007 for input to the AEG system. The typists are instructed to carefully preserve all the errors during the transcription process. These Microsoft Word documents serve as input to our AEG system.

As shown in figure 4.1, the conceptual framework consists of two main processes – the pre-processing stage called the filter process and the actual essay grading process. The input to the AEG system is the essay which is to be assessed and the output from the AEG system are the final scores from each module shown as the essay grade. Each process is explained in detail as follows.

### 4.2.1. Pre-processing stage- Filter process

A thorough manual examination of the essay dataset revealed that essays contained several types of improperly constructed responses. Some primary school students drew figures as a response while others drew figures and wrote a list of words related to the figure. Since primary school students of Year 3 are still learning that words are a string of letters, some students wrote a string of letters which did not make any sense at all as they were not proper words. Other students copied the question prompt as is and submitted it as a response. Moreover, since we are dealing with student essays, it is natural that there are spelling and

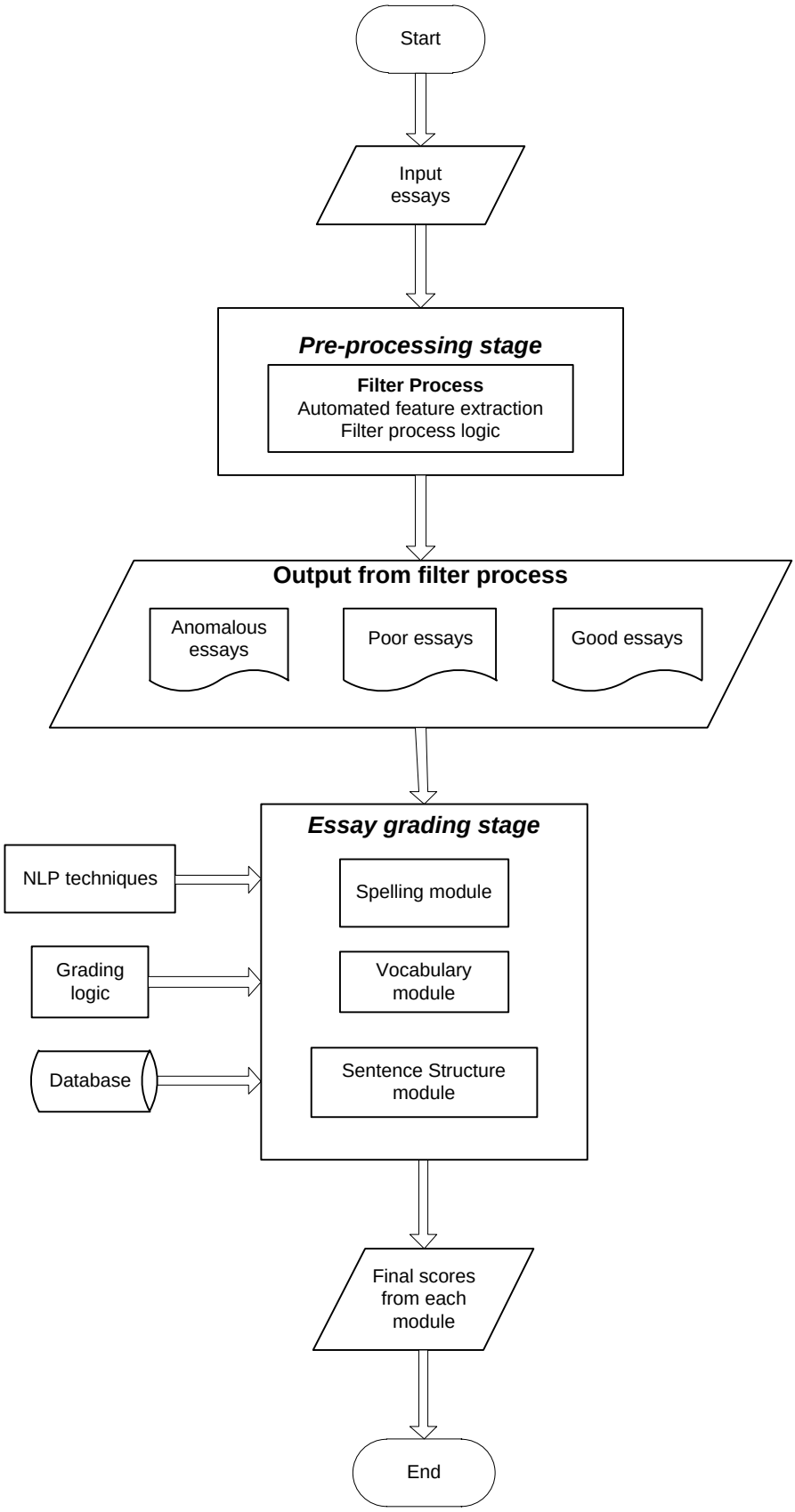



**Figure 4.6: Overview of the conceptual framework of the AEG system**

grammar errors. In fact, our manual examination revealed that some essays had so many mistakes that it was an arduous task to decipher what the student intended to say. Such essays are referred to as ‘noisy essays’. Futagi [3] attributes the misspellings, context-based spelling errors, morphological, or syntax-based errors and punctuation errors in the essays to 'noise'. One of his key findings is that, if an AEG system is built without initially filtering the noise from the essays, then it has a detrimental effect on the performance of the system. ‘Noisy’ essays in the dataset can potentially hamper and run down the proper performance of an AEG system. It is important to detect noisy essays because an AEG system that has been developed to assess and grade essays would throw unexpected errors if it encountered a ‘noisy’ essay. Such an AEG system would also grade a properly constructed response incorrectly because of the fault incorporated while grading the noisy essay. So it is important to detect them and eliminate them from the dataset. Hence, it is imperative to have a filter process which can filter out noisy essays from the dataset to enable the appropriate processing of essays.

Furthermore, the subtle assumption in PEG and LSA that an unscored essay reflects a properly constructed response [4] does not hold true in the case of our AEG system. In fact, our system is designed to grade any type of essay within the narrative genre. This is because we have a filter process in our AEG system and the purpose of this filter process is to filter out essays that are not properly constructed responses, as we will see further in this section.

Since we are trying to develop an automated system, we need to extract some features from the essays in order to develop the logic for the filter process. Keeping in mind our aim of maximising the use of available resources rather than building or creating new resources, we developed a program that can automatically obtain the errors detected by Microsoft Word 2007. We used Word 2007 because the essays are available as Word documents and for several other reasons as follows. Moreover, we choose Word because according to the latest estimates, it is ubiquitously present on millions of laptops, desktops and other devices and is by

far the most commonly used word processing software [5-7]. Further, [8] reports that students felt very comfortable using Word because they have access to it both at school and at home. Finally, Word incorporates a series of lexical, syntactic and semantic NLP tasks to perform contextual spelling and grammar error detection, as claimed by Microsoft Research [5] .

Microsoft Word 2007 incorporates a widely used spelling and grammar checker. Several studies have been undertaken on the performance and evaluation of the Word program. It is reported that the precision of Microsoft Word in error detection is remarkable [9]. However, the recall is lower which means that it responds to fewer error types. These findings are substantiated by the conclusions drawn by [10]. A thorough critical evaluation of the spell checker incorporated in Word 2007 reveals that although it fails to find some errors, when it does flag a possible error, it is almost always correct. Further, it has been reported that it is capable of detecting errors which are real-word errors (correctly spelled word but used out of context) as well as those which are non-word errors (incorrectly spelled words). Word can also detect errors in compound word formations such as “through out” for “throughout” and errors in usage of apostrophes, for instance, “theirs” for “their’s”. Despite the fact that Word has a low recall, the performance of Word 2007 is sufficient for our requirements for the initial pre-processing stage.

In the pre-processing stage, for every input essay, we extracted only four features from Microsoft Word and used them to build the logic for the filter process. The four features are: number of spelling errors, number of grammar errors, number of paragraphs and total number of words in the essay. Then, using a heuristics and rules-based approach, we developed the methodology for the filter process in order to detect essays that are improperly constructed

responses and ‘noisy’. We divided them into two types: anomalous essays and poor essays[1]. We define these two types of essays as follows.

1. Anomalous essay. An ‘anomalous’ essay can be defined as an essay that is :
    a. a blank response- the student has submitted a blank paper instead of writing an essay
    b. a picture – the student has drawn a figure instead of writing an essay
    c. a picture and a list of words – the student has drawn a figure and written a list of words related to the figure. However, there are no proper sentences in the response at all.
    d. question prompt as answer response – the student has copied the NAPLAN question prompt either as part of his response or completely as his/her response.
    e. written completely in upper-case letters – the student has written completely in upper-case letters. It is important to detect this type of essay because Word cannot properly detect the errors in such a document.
2. Poor essays. A 'poor' essay can be defined as an essay that is :
    a. gobbledygook - mainly random typing, which does not make any sense at all as there are no proper words in it.
    b. extremely poor in spelling and punctuation.
    c. too small to be called an essay - the essay consists of only a few words or sentence fragments.

[1] Details about our automated program for extraction of features and the complete logic for filter process in elucidated in the next chapter.

All essays that are neither anomalous nor poor are categorised as good essays. Hence, the output from the filter process is three datasets: one for anomalous essays, one for poor essays and another for good essays.

In the next section, the essay grading stage of the system is elucidated.

### 4.2.2. Essay grading stage

In this stage, three different modules are employed to score the essays in the criteria of spelling, vocabulary and sentence structure. These modules have access to a database specific to each, the logic for each module and natural language processing techniques as required. For some modules where we plan to use the neural network, it is essential that we develop different logic for grading poor essays and for grading good essays. This is to avoid the draw-backs of using the same logic for both types of essays, as pointed out in Section 4.2.1. In fact, the purpose of detecting anomalous and poor essays in the filter process is to eliminate their effect on the overall grading process and to grade them separately. The details of each module are explained as follows. The final scores obtained from these modules are displayed to the user as a result of the essay grading process.

It is important to note that these modules can run in parallel hence optimising the available computational resources of memory storage and processor requirements. This also results in the essay grading process being carried out in a matter of a few seconds per essay.

In the next section, the overview of the spelling module is given.

#### 4.2.2.1. Overview of ‘Spelling’ module

As stated in the dictionary, the spelling of a word is the correct order of letters. In order to assess spelling in an essay, it is essential to firstly identify the correct and incorrect words. Then, using the set of NAPLAN rules and guidelines for scoring the spelling criterion, the

score can be assigned. For this purpose, our spelling algorithm is based on some rules and heuristics. This method of designing an algorithm is supported by [11], where they used rules and heuristics to design the logical hyphenation program.

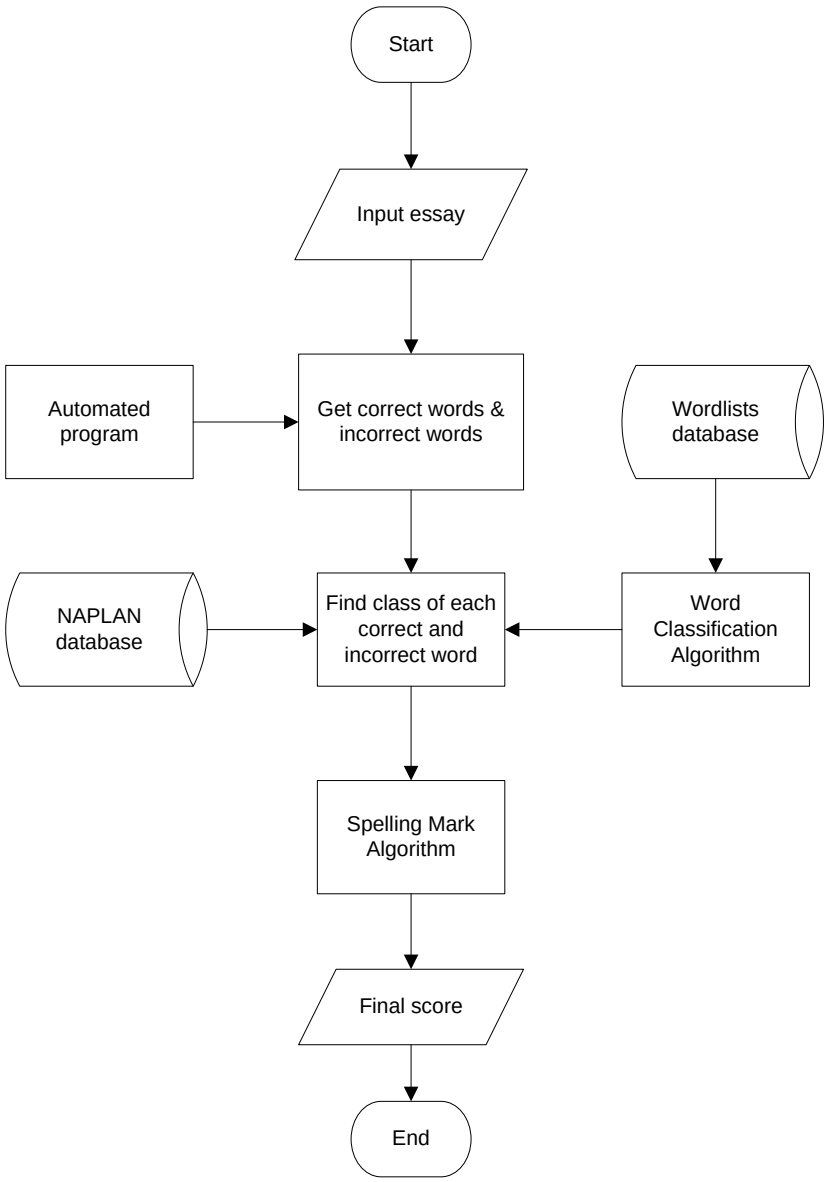



**Figure 4.7: Conceptual framework of the 'Spelling' module**

According to the NAPLAN guidelines, the skill focus for scoring spelling is 'the accuracy of spelling and the difficulty of words used'. Clearly, there are two measures to be assessed for the skill focus. Part 'A' checks the accuracy of spelling, for which we need to find the correct and incorrect words in the essay. As mentioned above, we use the context-sensitive spell

checking ability of Microsoft Word 2007 to find the correct and incorrect words. Word underlines incorrectly spelled words with red lines and gives a list of suggestions of correct spellings for each word. Using the automated program called 'SpellChecker' that we developed above; we extract the incorrect words and the first suggestion for each word. All other words are treated as correct words. Armed with this information, we move onto the next phase in the module.

Part 'B' assesses the difficulty of words used. For this purpose, NAPLAN classifies words into one of the four classes: simple, common, difficult and challenging. This classification is done based on the decreasing ease in the spelling of the word. Some examples of each class are given in the previous chapter in Table 3.2. NAPLAN provides a database of 1196 words to serve as a guideline that is used by human markers in conjunction with the several criteria stated. The size of this database is given in Table 4.1 and the complete list of words is given in the NAPLAN marking guide [12].

**Table 4.16: NAPLAN criteria for word classification**

| NAPLAN classification | Total words in database | Example criteria |
|---|---|---|
| Simple | 174 | a single syllable word with double final consonants |
| Common | 432 | single syllable words ending in *ould, ey, ough* |
| Difficult | 398 | multisyllabic words ending in *tion, si-on, ture* |
| Challenging | 192 | Foreign words |

In the classes - simple, common, difficult and challenging, there are 174, 432, 398 and 192 words in the NAPLAN database respectively. If a particular word is not found in the data-

base, then criteria are used to determine the class of a word. An example criterion to determine if a word can be classified as ‘simple’ is if the word has a single syllable and has double final consonants. An example criterion to determine if a word can be classified as ‘common’ is if the word has single syllable and ends in *‘ould’, ‘ey’*, or *‘ough’*. An example criterion to determine if a word can be classified as ‘difficult’ is if the word contains multiple syllables and ends in *‘tion’, ‘sion’,* or *‘ture’*. An example criterion to determine if a word can be classified as ‘challenging’ is if it is a foreign word.

In order to formalise these criteria, the word classification algorithm is developed. It is explained in the next section.

#### 4.2.2.1.1. Word classification algorithm

There are various explicitly given criteria, based on which a word is classified into one of the four classes. We formalise these criteria and develop the ‘Word Classification algorithm’ to automatically classify a new word. An understanding of the following terms serves as a foundation to comprehend the complete algorithm.

**Consonants**. The complete alphabet in English language, except the vowels (a, e, i, o, u) are called consonants.

**Consonant digraphs**. These consist of two consonants that make one sound when blended. Ex: ch, ph, th, wh, etc. Using several English language teaching resources, we compiled a list of consonant digraphs for the purpose of our algorithm.

**Consonant blends**. Consonant blends can be either di-blends or tri-blends.

Consonant di-blends consist of two consonants that make two distinct sounds when pronounced, for example: br, ft, ld, etc. The only difference between di-blends and di-graphs is

that di-graphs do not retain the sound of each of the consonants involved. Most often, they produce a new sound.

Consonant tri-blends consist of three consonants that make three distinct sounds when pronounced, for example: str, tch, nth, phr, etc. Using several English language teaching resources, we compiled a list of consonant di-blends and tri-blends for the purpose of our algorithm.

**Syllable**. A syllable is 'one or more letters representing a unit of spoken language consisting of a single uninterrupted sound'. A word with one syllable is called a mono-syllabic or single syllabic word, for example: 'bun', 'one', 'I', 'all'. A word with two syllables is called di-syllabic, for example: 'body' (bo_dy). A word with three syllables is called tri-syllabic, for example, 'anyone' (a_ny_one). A word with more than three syllables is called multisyllabic, for example: 'anybody' has four syllables (a_ny_bo_dy) and 'responsibility' has six syllables (res_pon_si_bi_li_ty). To find the number of syllables in each word, we use the database provided by [35]. This database consists of about 50,000 words ranging from simple to challenging classes. This database is chosen because it satisfies our requirements for building the word classification algorithm according to NAPLAN guidelines.

**Short vowel and Long vowel**. According to the orthographic pronunciation rules of the English language, a short vowel is when the pronunciation of the vowel is short (ex: hot) and a long vowel is when the vowel is pronounced at length (ex: bait). The length of the vowel is denoted using special characters from the International Phonetic Alphabet (IPA) [13]. A close examination of words with short vowel and long vowel pronunciations bring to light that most often long vowels are words that have more than one vowel. Hence for the purpose of this thesis  and since we do not have access to an IPA dictionary, we define a *short vowel*

*sound* as a word that has only one vowel and a *long vowel sound* as a word that has more than one vowel. This rule is to be used only in the context of this program.

**Homophones**. These are words that are pronounced alike but differ in spelling and meaning.There are hundreds of homophones in the English language. The most common homophone errors are it's/its, to/too/two, there/their/they're, who's/whose, weather/whether, lose/loose, where/were, past/passed, principle/principal and quiet/quite [14]. For the purpose of this thesis, we use the homophones list available at [15, 16]. There are about 2000 pairs of homophones in the compiled lists.

**Compound words**. When two different words are joined to make one word, it is called a compound word. The compound word does not necessarily have the same meaning as the individual words themselves, for example: understand (under + stand), screwdriver (screw + driver). There are over 2000 compound words in the English language. Compound words do not have a space between them. For the purpose of this thesis, we used the freely available compound word lists at [17, 18]. There are about 5000 words in the compiled lists.

**Affix**. An affix is a smallest semantically meaningful unit that is attached to a word stem to form a new word. An affix can be either a prefix or suffix, for example: '-less' in baseless, 'un-' in undo. We use the 10 most productive affixes used to form adjectives, adverbs, nouns and verbs, as reported in [19]. We find that this list of affixes satisfies most of our requirements.

**Contractions**. A contraction is a shortened form of a word or group of words, with the missing letters usually marked by an apostrophe for example: let's, shouldn't, I'll. For the purpose of this thesis, we compile a list of contractions from freely available sources such as teaching resources for teaching English. There are about 50 words in this list.

All the above constitutes the database for the word classification algorithm. The logic of the algorithm is based on certain heuristics and rules coded in Java. This method of designing an algorithm is supported by [11], where they used rules and heuristics to design a logical hyphenation program. This method of designing an algorithm is analogous to the work in [20].

Based on the NAPLAN rubric guidelines, the algorithm is designed. For a word to be classified, we take into consideration two specific attributes unique to the word, its word length and the number of syllables. The length of a word is the number of characters in the word which can be easily counted by the program. All the words with word length of 1 or 2 are classified as simple. NAPLAN dictates that all contractions are to be classified as common. Hence, if the word is found in the list of contractions then it is classified as common. For all words with length greater than 3, the number of syllables in the word is determined by looking up in the syllables dictionary.

- If the word is mono-syllabic, then we check if it has a consonant digraph or a consonant di-blend or double final consonants or short vowel to classify the word as simple. If the word has two consonant digraphs or two consonant di-blends or consonant tri-blend or a combination of any of these or long vowel, then we classify the word as common. If a word ends in *-ough*, *-ey*, *-ught* it is classified as common. If a word has an affix such that the affix does not change the base word, the word is classified as common, else it will be difficult.

  In Natural Language Processing (NLP), the base word can be derived from a word by using either a stemmer or a lemmatiser. A stemmer is a program that roughly chops off the suffix from the word and based on certain rules, returns the base word. It is a very rudimentary tool which does no more than suffix stripping. A more refined and better performance is obtained by using a lemmatiser, which is a program that takes into consideration

the part of speech of the word and using complete morphological analysis, returns the base word or the dictionary form of the word, called the lemma. Both these techniques cannot be used in the context of our program of Word classification for a number of reasons. Firstly, NAPLAN is concerned with the base word but not the stem of the word. Hence, we cannot use the stemmer. Moreover, since NAPLAN does not provide the part of speech of the word, we cannot use the lemmatiser. As an alternative to both these methods, we develop code that simply removes the affix from the word and if the remaining word is found in the dictionary lookup, then it is returned as the base word. This simple logic serves the purpose as described by NAPLAN.

Hence, a mono-syllabic word can be simple, common or difficult. If the word is bi-syllabic or tri-syllabic, then depending on its word length and if it is found in the compound list or homophones list, it is either common or difficult. If the word has the suffix '-ing' then depending on whether it has a prefix or not, it is either difficult or common. If word has only one affix then depending on the word length of the base word, it is either common or difficult. Hence, a di-syllabic or tri-syllabic word can be either common or difficult.

- If the word is multi-syllabic, and if it is found in the compound word list, then it is classified as difficult. If the word ends in *tion, ent/ant, ful, ture, ible/able, sion,* then it is difficult. If the word has an affix, then depending on the length of the base word, it is either difficult or challenging. If the word has a suffix and the base word ends in e, c or l then it is challenging. Multisyllabic words that do not satisfy any of these rules are automatically classified as challenging. Hence, a multi-syllabic word can be either difficult or challenging.

In the next section, the spelling mark algorithm is described. This algorithm is used for automatically assigning the spelling score to the essay.

#### 4.2.2.1.2. Spelling Mark Algorithm

After finding the class of each correct word and the class of the first suggestion of every incorrect word, we move onto the next phase and feed these inputs into the spelling mark algorithm. This algorithm is capable of assigning the spelling score to the essay. To develop this algorithm, we formalised the NAPLAN guidelines for each score within the score band of 0-6. According to the guidelines, if the essay contains a figure or no proper evidence of conventional spelling, then it is assigned a score '0'. In order to achieve this, it is taken that the number of words in the essay is equal to the number of spelling errors in the essay. On the other hand, if the number of words in the essay is not equal to the number of spelling errors, then we do the following. If there are only simple words in the essay, it will receive a score of '1'. If the correct words are simple and common but the incorrect common words are less than the number of correct common words, it will receive a score of '2'. For a score of '3', the essay should contain most simple words and at least 20 common words which are spelt correctly. For this category, it suffices if there are at least 20 common words, irrespective of the errors.

On the other hand, for a score of '4', the essay should have all the simple words, most common words and at least 2 difficult words spelt correctly, but the number of incorrect difficult words should be less than the number of correct common words. For a score of '5', all simple words and most common words should be correct. Additionally, for this score category, at least 10 correct difficult words should be present in the essay and the number of incorrect difficult words should be less than the number of correct difficult words. For the highest score i.e., score '6' in this criterion, the essay should have correct spelling of all simple and common words, at least 10 difficult words and at least 2 challenging words. If there are no challenging words, then at least 15 correct difficult words are required with the condition that

there can be only 1 or 2 minor occasional errors. If there are any more errors, then the essay cannot be assigned to this category, and will have to be assigned a score of '5'. Using heuristics and the above rules, we developed the spelling mark algorithm to portray the above formalisation. The spelling mark algorithm produces the final spelling score for the essay as the output, which is displayed to the end user.

The next section provides an overview of the vocabulary module which is responsible for assessing an essay and assigning a vocabulary score automatically to the essay.

#### 4.2.2.2. Overview of 'Vocabulary' module

A person with a good vocabulary is able to use a variety of words to convey their meaning and expression. In order to assess vocabulary in an essay, it is essential to assess the range of words used and the appropriateness of their usage. The words being used can range from simple 2 or 3 letter words to difficult or challenging words such as technical terms (for example, gynaecology). According to the NAPLAN rubric, the skill focus in assessing vocabulary is the range and precision of language choices. In order to formalise the rubric guidelines, we propose that *range* can be attributed to the quantity of lexical items used where according to NAPLAN; the lexical items can be either noun, verb, adverb, adjective, noun groups, phrasal verbs or verb groups. The definitions of these terms are given in chapter 3, section 3.7.2.2. More specifically, *range* can be measured by finding the number of unique words in each lexical item.

Our understanding of the rubric guidelines maintains that p*recision* can be attributed to the quality of the words. The quality of the words can be measured by taking into consideration the class of the word: simple, common, difficult or challenging.

As per the NAPLAN rubric guidelines, the vocabulary score lies within the score band of 0-5. To develop the vocabulary module, we divide the score band into two categories: a score of 0-2 for poor essays and a score of 3-5 for good essays. This division is essential because there is a substantial shift evident in the level of vocabulary from a score of 0-2 to a score of 3-5. For example, an essay with a score of 2 would have mostly simple lexical items such as *quick, big, water* whereas an essay with a score of 3 would have mostly precise words such as *hissed, yanked,* and *clutched*. Poor essays and good essays are datasets obtained from the filter process that was carried out in the pre-processing stage of the AEG system, explained in detail in section 4.2.1. It is important to develop separate algorithms for both types of essays because of the inherent differences between the qualities of essays that fall in each type. Poor essays are those which have excessive spelling and grammatical errors to the extent that they seem gobbledygook, as shown below in example.

*in a tugs was ag, I goatady. Idhe was t redia no, atso. So was bafso bo we a no sutatu horse sle wansue, oettace I laso tudiaesol soogs insipiia wetue scana.*

**Figure 4.3: Example of a gobbledygook essay**

Essays which have fewer than 80 words are also categorised as poor essays. Moreover, poor essays have very rudimentary sentence constructions, if at all there are any sentences. Hence, this would lead to incorrect conclusions if we use a higher level of NLP techniques to analyse the range and precision. Taking into consideration the aforesaid reasons, we develop an algorithm for scoring vocabulary for poor essays.

In order to grade vocabulary, two things are of main importance: (1) content words-the number of unique words in classes nouns, verbs, adverbs, adjectives, noun groups, phrasal verbs and verb groups; and (2) grammatical word classes-the number of unique words in classes prepositions, articles, conjunctions, pronouns and interjections. So, in essence, we are looking

for variety (the more, the better), quality (the more precise, the better) and quantity (the more, the better proof of good vocabulary) of different classes of words.

Even though the essay might have repetitive words to try to "fool" the system into giving a higher score for vocabulary, our program will collect only unique words in each class. So, if the essay has repetitive words, the algorithm will consider each word only once.

According to [23], counts of part of speech tags are better indicators of the complexity of the text, when compared to only surface features or parse features. The results that were reported explain that the most predictive features for document classification were number of adjectives (more for complexity), among others related to sentence structure. We use these features and apply them to essay grading in order to calculate the vocabulary grade of the essay.

#### 4.2.2.2.1. Algorithm for poor essays

According to the NAPLAN rubric, if an essay has symbols or drawings, it is assigned a score of '0'. This is easy to achieve and we assign this score category to anomalous essays case 1, detected in the filter process. Poor essays have basic vocabulary. The rubric states that if an essay has a very short script, then it is to be assigned a score of '1'. If an essay has mostly simple lexical items, then it is to be assigned a score of '2'. In order to achieve this, we compile a database consisting of basic vocabulary words from two widely recognised word lists, the Voice of America Special English word list [21] and the Ogden Basic English list [22]. Both these word lists consist of basic vocabulary items. These word lists are widely used in research related to text simplification processes, as evident from the work in [23]. The Voice of America Special English word list consists of about 1500 words compiled from the Special English programs on radio, television and the Internet. The Ogden Basic English list consists of 850 words, of which there are 400 general nouns, 100 adjectives, 100 verb-forms (opera-

tors), articles, etc., 200 judiciously selected names of picture-able objects (common things such as the parts of the body), and 50 adjectival opposites.

In the logic for poor essays as illustrated in figure 4.4 - for an essay, based on the percentage of words found in the database, the vocabulary score is assigned to either a score of '1' or a score of '2'. If the percentage is 0, then the vocabulary score is assigned as 0. If the percentage is below a certain threshold, then the vocabulary score is assigned as 1. If the percentage is above a certain threshold but the total number of words in the essay is less than 30, then

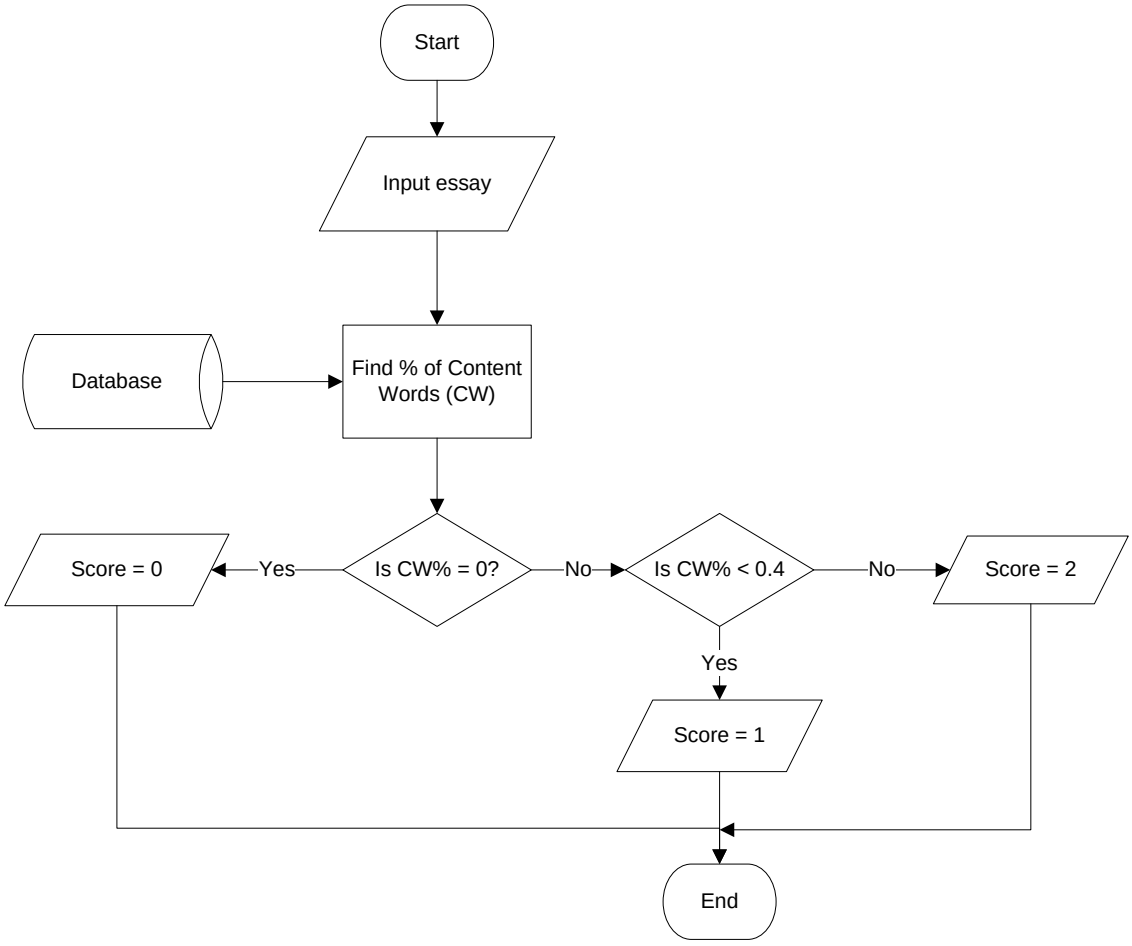



**Figure 4.4: Conceptual framework for grading vocabulary in poor essays**

again, the vocabulary score is set to 1. This is because according to NAPLAN, if an essay has only a few content words, then it is assigned a score of 1. The number 30 is derived empirically. Otherwise the vocabulary score is assigned as 2.

#### 4.2.2.2.2. Algorithm for good essays

In NLP, Part Of Speech (POS) tagging is a technique used to identify the different parts of speech in a sentence. It takes a sentence as input and assigns the parts of speech such as noun, verb, adjective, etc. to each word. The advantage of using a POS tagger is that it is not affected by incorrect spellings. Table 4.2 provides more information on the output obtained from a part-of-speech tagger.

**Table 4.17: Description of Penn Treebank tags produced by the POS tagger**

| Part of Speech Tags | Examples | Example sentence |
|---|---|---|
| Determiner (DT) | An, another, any, both | **The** big bus<br><br>Take **all** apples |
| Adjective (ADJ) | Cold, bright, sharpest | The **big** bus<br><br>Take all **red** apples |
| Noun (NN) | Human, pencil, wind | The big **bus**<br><br>Take all **apples** |
| Adverb (RB) | Healthier, further, swiftly | She ran **fast**<br><br>He went **later** |
| Verb (VB) | Eat, give, soaked | She **ran** fast<br><br>He **went** later |
| Pronoun (PRP) | Herself, they, your | **She** ran fast |

| | | |
|---|---|---|
| | | **He** went later |
| WH-pronoun (WP) | Which, what, who | With **whom** shall I meet? |
| Subordinating Preposition or Conjunction (IN) | Outside, below, against | **Inside** the house<br><br>She stood **near** them |

Table 4.2 shows different parts of speech tags that are assigned by a POS tagger to the words in the sample sentence.

We use the freely downloadable Stanford Log-linear part-of-speech tagger [24] because it is reported to be accurate up to 97.5% and is in the Java language. It takes a sentence as input and provides the part of speech of each word in the sentence as output. It uses the tags from the Penn Treebank tag set, available here [25].

*Sentence* : I looked up from the book I was reading and saw two people walking towards me.

*POS tagged output* : I/PRP looked/VBD up/RP from/IN the/DT book/NN I/PRP was/VBD reading/VBG and/CC saw/VBD two/CD people/NNS walking/VBG towards/IN me/PRP ./.

**Figure 4.5: Sample output from Stanford POS tagger**

We processed an essay through the tagger and obtained all the tags. Since the tagger provides different tags for different forms of verbs, we use coarse tags wherein we combine similar tags into one category. For example, in the output provided by the tagger, the POS tag for the base form of a verb is VB but the POS tag for the past tense of a verb is VBD, for the present participle of a verb is VBG, for the past participle of a verb is VBN, for the present tense of the verb but not the 3$^{rd}$ person singular form is VBP and for the present tense of a verb and the 3$^{rd}$ person singular form is VBZ. Hence, the tags for all different forms of verbs will be combined into the course tag of 'verb'. We adapt the coarse tags from [23]. Then, we determine the number of unique words in each coarse tag.

From the essay, we find the number of unique words in each class, Simple, Common, Difficult and Challenging. We use the NAPLAN database on word classification to find the class of the words.

We find the frequency of words in each class. We use frequency because if a student has a limited vocabulary, he will use the same words repeatedly. Hence, the frequency will be higher. On the other hand, if a student has a good vocabulary, he will use a variety of words. For example, a student might use the same word ‘car’ to refer to a car in his essay whereas a student whose vocabulary is better might use ‘car, vehicle, automobile’ to refer to the car.

We also use some general features of the essay such as total number of words. We use general vocabulary measures such as average word length [26], Flesch readability ease and Flesch K-grade level. These measures of vocabulary have been widely recognised and used since Flesch proposed them in 1948 [27]. Since MS Word provides these measures, we obtain these values automatically from MS Word using our 'Spellchecker' program. We use all these as inputs to the neural network and carry out the simulations.

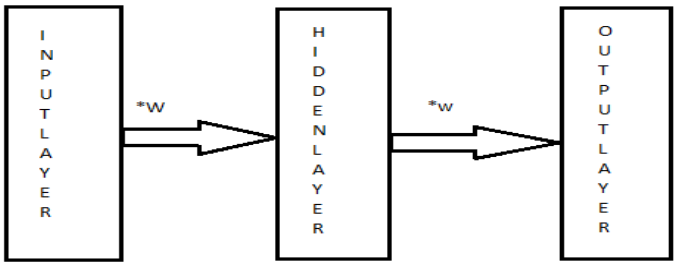



**Figure 4.6: Simple design of a neural network (from [28])**

An Artificial Neural Network (ANN) is a machine learning technique. To the best of our knowledge, it has been used by only one AEG system so far.

Figure 4.6 shows the design of a multi-layer feed forward neural network (MLFFN) and uses the back propagation algorithm as the learning algorithm [28, 29]. This neural network is able to model non-linear relationships between the inputs and outputs. It has three layers: the *Input layer* consists of input nodes; the *Hidden layer* consists of hidden layer nodes; and the *Output layer* consists of output node. The inputs to the neural network are fed through the input nodes. MLFFN is most commonly used for prediction, pattern recognition, and nonlinear function fitting.

We feed the various inputs, described above, through the input nodes to the hidden layer. The inter connections between the nodes in the various layers have weights assigned to them, which the neural network uses to 'learn'. Processing and computations from inputs are carried out in the hidden layer by using the weights and the inputs. The final grade for vocabulary is obtained as an output from the neural network.

In the next section, an overview of the sentence structure module is presented.

#### 4.2.2.3. Overview of 'Sentence Structure' module

According to the NAPLAN rubric, the skill focus in assessing sentence structure is "the production of grammatically correct, structurally sound and meaningful sentences". It is scored within a band of 0-6. A grammatically incorrect sentence has an error violating one or more rules of English grammar [30]. Identifying and correcting grammatical mistakes is an ongoing research topic which is not possible within the scope of this project. Since our aim is to maximise the use of available resources, we use the grammar checking abilities of MS Word to detect poor essays and any essay that is not detected as a poor essay will have sentences that

are readable and have few errors. Hence, we scan them through the parser and extract other features that determine their grade. It should be noted here that the logical meaning or the semantic correctness of sentences cannot be checked by the system at this stage.

While students from primary school are still learning to form simple sentences, students from secondary school will have mastered the constructions of simple sentences, have learned compound and complex sentence formations and will mostly be using a variety of all formations. Keeping in mind the reasons mentioned in previous section, we develop two different algorithms: one for scoring poor essays and the other for scoring good essays. For this purpose, we divide the score band into two: a score of 0-3 for poor essays and a score of 4-6 for good essays. Poor essays and good essays are datasets obtained from the filter process and are carried out in the pre-processing stage of the AEG system, as explained in detail in section 4.2.1. This division is essential because there is a substantial shift evident in the sentence structure formations and a variety of scores from a score of 0-3 to a score of 4-6. It is important to develop separate algorithms for both types of essays because of the inherent differences between the qualities of essays that fall in each type.

In the next section, the methodology of algorithm for grading sentence structure in poor essays is described.

#### 4.2.2.3.1. Algorithm for Poor essays

According to the NAPLAN rubric, if the essay has figures or a list of words or only fragments of sentences, then it is to be assigned a score of '0'. Hence, we assign this score to all the anomalous essays except case 5. If the essay has only one sentence, then we run it through the chunker.

In NLP, chunking is a technique used to divide a sentence into syntactically related non-overlapping groups of words. These groups of words are called phrases. The chunker is also

called a shallow parser and was originally built to be an aid to the full parsing of sentences. The phrase chunking technique lies between POS tagging and a comprehensive grammar analysis (parsing) [31]. For a POS tagged text, phrase chunking attaches tags for noun phrases, verb phrases, etc. The output from the chunker consists of phrase tags such as NP for noun phrases, VP for verb phrases, ADJP for adjectival phrases and ADVP for adverbial phrases. Table 4.3 provides more information on the various types of phrases in the output obtained from a chunker.

**Table 4.18: Description of Penn Treebank Phrase tags produced by the Chunker**

| Phrase tag | Description | Example |
|---|---|---|
| NP | Noun Phrase | **The boy** was gone.<br><br>It was **pitch dark**. |
| VP | Verb Phrase | The boy **was gone**.<br><br>It **was** pitch dark. |
| ADJP | Adjectival Phrase | The slope was getting **steep-er**.<br><br>Steve moved **closer and closer**. |
| ADVP | Adverbial Phrase | Ryan had **just** lost his wallet.<br><br>The ship was steered **safely and smoothly.** |

| | | |
|---|---|---|
| SBAR | Clause introduced by subordinating conjunction | She treated us **as if** she liked us.<br><br>**Inspite of** the rain, the game was on. |
| PP | Prepositional Phrase | A breeze **through** my chest.<br><br>It slipped **between** the rocks. |

A chunker takes a sentence as input and assigns the different phrase tags in the sentence as output. The advantage of using a chunker is that it is not affected by incorrect spellings. An example sentence and its corresponding output obtained from a chunker are given in the figure 4.7 below.

*Sentence* : The effort of shoving snow into the cave was overwhelming.

*Output from Chunker:* [NP (DT The) (NN effort) ] [PP (IN of) ] [VP (VBG shoving) ] [NP (NN snow) ] [PP (IN into) ] [NP (DT the) (NN cave) ] [VP (VBD was) ] [ADJP (JJ overwhelming) ] (. .)

**Figure 4.7: Sample output from Illinois Chunker**

The square brackets in the 'Output from Chunker' enclose a tag and its constituents. The tag-name is at the head of the square brackets. The tag constituents are words and their part of speech tags. We use the freely downloadable Illinois Chunker because it is reported to be very accurate, is in the Java language and is widely used in text mining [32].

For other essays, we need to determine the type of sentences present and whether each sentence is correct or incorrect. To identify the type of each sentence, we determine if it is a simple, compound or complex sentence. We use the linking and binding conjunctions lists provided with the NAPLAN guide for this purpose. As mentioned earlier, the linking conjunctions are used in compound sentences and the binding conjunctions are used in binding con-

junctions. Using keyword matching, we first check for the presence of a binding conjunction. If it is present, then the sentence is classified as a complex sentence. If not, then we check for the presence of a linking conjunction to classify the sentence as a compound sentence. Otherwise, the sentence is regarded as a simple sentence.

To determine the correctness of a sentence, we check if the sentence has a grammatical error in the output obtained from our 'SpellChecker program'. Then, using the mathematical formulation and depending on the percentage of correct simple sentences, the percentage of correct compound sentences and the number of correct complex sentences, the scores for sentence structure are assigned.

#### 4.2.2.3.2. Algorithm for Good essays

A careful manual examination enables us to conclude that good essays generally have a variety of sentence structures ranging from simple sentences to complex sentences. This is further supported by the rubric guidelines that state that for an essay to achieve a score greater than 3, it should demonstrate a *variety* of sentence structures and should use both basic and sophisticated structures. Our hypothesis is that if we could capture the types of sentence structures and analyse them, we can grade them. As mentioned earlier, a chunker can detect the various phrases in a sentence. While doing so, it detects subordinate clauses as well, which are present in compound and complex sentences. To capture the types of sentence structures, we first scan them through the chunker and extract the various types of phrases.

According to Chomsky's phrase structure grammar, a simple sentence consists of a NP and a VP [33]. As the complexity of the sentence increases, the number of phrases in the sentence increase. So ADJP, ADVP and SBAR phrases are present in compound and complex sentences. Based on this, we extract the syntactic structures of the sentences, as shown effective in [34], which reports on the various features that influence sentence fluency. Some of the fea-

tures that we employ for essay grade prediction are average sentence length, number of subordinating conjunctions (SBAR count including SBARQ), number of noun phrases (NPs), number of verb phrases (VPs), number of prepositional phrases (PPs), number of adjectival phrases (ADJPs) and number of adverbial phrases (ADVPs). We also determine the length of the noun phrase because their work shows that as the length of noun phrase increases, the complexity of the sentence increases. We use a multi-layer feed forward neural network with back propagation algorithm as the learning algorithm (explained in detail in section 4.2.2.2.4) to compute the sentence structure score for the essay.

In the next section, the main points of this chapter are recapped to serve as conclusion.

## 4.3. Conclusion

In this chapter, the conceptual framework of the proposed AEG system was explained, showing that it consists of two main stages: the pre-processing stage and the essay grading stage. The pre-processing stage comprises the filter process. As our aim is to maximise the use of available resources, we use the spelling and grammar checking abilities of Microsoft Word to extract certain features from the essay to serve as input for the filter process. As an output from the filter process, anomalous essays and poor essays were obtained. All the essays which are neither anomalous nor poor are considered to be good essays. The output generated was sufficient according to our requirements and allowed us to progress to the next stage in the AEG system, which is the essay grading stage.

The essay grading stage is where the actual, in-depth processing of essays takes place. It comprises three separate modules, one each for grading spelling, vocabulary and sentence structure, according to the NAPLAN rubric. In the description of the spelling module, the conceptual framework of the word classification algorithm and the spelling mark algorithm were explained. In the description of the vocabulary module, two separate algorithms, one for grading poor essays and another for grading good essays were described. In the description of the sentence structure module, the conceptual framework of the algorithms for grading poor essays and good essays separately was detailed.

The next chapter will explain the working of the filter process in greater detail and presents a preliminary data analysis of our dataset. Then, we perform a preliminary neural network simulation to study the feasibility of using the neural network for grading essays and to perform feature optimization.

## 4.4. References


[1] N. A. M. Razali, *et al.*, "Heuristics and Rule-based Approach for Automated Marking Tool for ESL writing," in *Proceedings of International Symposium of Information Technology (ITSIM)*, 2008, pp. 144-149.

[2] N. Omar, "Heuristics-based Entity-Relationship Modelling through Natural Language Processing," Ph.D, Faculty of Engineering, University of Ulster, 2004.

[3] Y. Futagi, "The effects of learner errors on the development of a collocation detection tool," presented at the Fourth workshop on Analytics for noisy unstructured text data, Toronto, ON, Canada, 2010.

[4] K. W. K. Chung and H. F. O'Neil, "Methodological approaches to online scoring of essays," University of California, Los Angeles, Technical Report no. 461, 1997.

[5] T. McGee and P. Ericsson, "The politics of the program: MS WORD as the invisible grammarian," *Computers and Composition*, vol. 19, pp. 453-470, 2002.

[6] Microsoft. (2009, 30 October, 2011). Microsoft Office is Right at Home.

[7] Wikipedia. (2011, 30 October, 2011). Word Processor. Available: http://en.wikipedia.org/wiki/Word_processor

[8] C.-F. E. Chen and W.-Y. E. Cheng, "Beyond the Design of Automated Writing Evaluation: Pedagogical Practices and Perceived Learning Effectiveness in EFL Writing Classes," *Language Learning & Technology*, vol. 12(2), pp. 94-112, 2008.

[9] M. Chodorow and C. Leacock, "An unsupervised method for detecting grammatical errors," presented at the Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, Seattle, Washington, 2000.

[10] G. Hirst. (2008). *An evaluation of the contextual spelling checker of Microsoft Office Word 2007*. Available: http://ftp.cs.toronto.edu/pub/gh/Hirst-2008-Word.pdf

[11] W. A. Ocker, "A Program to Hyphenate English Words," *IEEE Transactions on Engineering Writing and Speech*, vol. 14, pp. 53-59, 1971.

[12] Department of Education, *Narrative Marking Guide 2010, National Assessment Program-Literacy and Numeracy*: Government of Western Australia, 2010.

[13] (2011, 14 October). *Vowel Length*. Available: http://en.wikipedia.org/wiki/Vowel_length#Traditional_long_and_short_vowels_in_English_orthography

[14] H. M. Breland, "Word Frequency and Word Difficulty: A Comparison of Counts in Four Corpora," *Psychological Science,* vol. 7, pp. 96-99, 1996.

[15] I. Miller. (13 June 2011). *English Homophones.* Available: http://www.bifroest.demon.co.uk/misc/homophones-list.html

[16] E. Antworth. (15 June 2011). *Homophones based on William Cameron Townsend's book 'Handbook of Homophones', 1975*.

[17] R. Walton. (15 June 2011). *2276 Compound Words.* Available: http://www.rickwalton.com/curricul/compound.htm

[18] D. R. Cooper. (15 June 2011). *Compound Word List*. Available: http://www.learningdifferences.com/Main%20Page/Topics/Compound%20Word%20Lists/Compound_Word_%20Lists_complete.htm

[19] A. Neviarouskaya, *et al.*, "SentiFul: A Lexicon for Sentiment Analysis," *Affective Computing, IEEE Transactions on,* vol. 2, pp. 22-36, 2011.

[20] M. Miłkowski, "Developing an open-source, rule-based proofreading tool," *Software: Practice and Experience,* vol. 40, pp. 543-566, 2010.

[21] Voice of America. (2009, 5 August 2011). *Word Book (50th Anniversary ed.)*. Available: www.voaspecialenglish.com

[22] C. K. Ogden, *Basic English: A General Introduction with Rules and Grammar*: Paul Treber & Co., Ltd., 1930.

[23] C. Napoles and M. Dredze, "Learning simple Wikipedia: a cogitation in ascertaining abecedarian language," presented at the Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids, Los Angeles, California, 2010.

[24] K. Toutanova, *et al.*, "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network," in *Proceedings of HLT-NAACL,* 2003, pp. 252-259.

[25] M. P. Marcus, *et al.*, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics (Special Issue on Using Large Corpora)*, vol. 19(2), pp. 313-330, 1993.

[26] A. Ben-Simon and R. E. Bennett, "Toward More Substantively Meaningful Automated Essay Scoring," *Journal of Technology, Learning, and Assessment*, vol. 6(1), 2007.

[27] R. Flesch, "A new readability yardstick," *Journal of Applied Psychology*, vol. 32, pp. 221-233, 1948.

[28] T. S. Dillon and D. Niebur, *Neural networks applications in power systems*: CRL Publishing Ltd, Market Harborough, UK, 1996.

[29] D. E. Rumelhart and J. L. McClelland, *Parallel distributed processing: Explorations in the microstructure of cognition.* Cambridge,MA: MIT Press, 1986.

[30] N. K. Nikitas, "Computer Assisted Assessment (CAA) of Free-Text: Literature Review and the Specification of an Alternative CAA System," 2010, pp. 116-118.

[31] D. Naber, "A Rule Based Style and Grammar Checker," Masters thesis, Faculty of Computer Science, University of Bielefeld, 2003.

[32] A. Rozovskaya and D. Roth, "Training Paradigms for Correcting Errors in Grammar and Usage," in *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, Los Angeles, CA, 2010.

[33] N. Chomsky, "Three models for the description of language," *Information Theory, IRE Transactions on*, vol. 2, pp. 113-124, 1956.

[34] A. Nenkova, *et al.*, "Structural features for predicting the linguistic quality of text: applications to machine translation, automatic summarization and human-authored text," in *Empirical methods in natural language generation*, K. Emiel, *et al.*, Eds., ed: Springer-Verlag, 2010, pp. 222-241.

[35] Vineet Dwivedi. (2009, 28 October, 2011). www.beedictionary.com.

# Chapter 5: Preliminary Analysis

## 5.1. Introduction

In the previous chapter, the conceptual framework of the AEG system is illustrated and described. An overview of the filter process, which is the pre-processing stage in the AEG system, is presented.

In this chapter, the actual working of the filter process is explained. The custom-written program that is developed in order to extract features from Microsoft Word automatically is illustrated and explained, after which descriptive statistics of our dataset are presented. Finally, a preliminary neural network simulation is performed to study the feasibility of using the neural network to grade student essays. In doing so, the linear and non-linear relationships between the essay features and the final grade are modelled. Finally, feature optimization is performed and the influence of features on the final essay grade is reported.

In the next section, the pre-processing stage is explained in detail.

## 5.2. Pre-processing Stage

As mentioned in the previous chapter, the purpose of the pre-processing stage is to detect and separate 'noisy' essays from the essay dataset. If allowed to enter the grading phase with the other essays, 'noisy' essays in the dataset can potentially hamper and run down the proper performance of an AEG system. So it is important to detect them and eliminate them from the dataset. Consider the example of an anomalous essay in figure 5.1.

Your story might be about finding a lost pet, hidden treasures or new friends. It could be about finding the solution to a problem or finding an opportunity to do something different and exciting. Your story could be about how people in difficult situations find courage, help or understanding. Think about:

- The characters and where they are
- The complication or problem to be solved.
- How the story will end.

**Figure 5.1: Example of anomalous essay type 4.**

In the above example, the essay appears to be well-written with very few mistakes. However, on closer scrutiny, it comes to light that the essay is actually the NAPLAN question prompt which the student has naively or cleverly copied and produced as the answer response. In the filter process described in the next section, all 5 cases of anomalous essays and poor essays are filtered out from the essay dataset.

In the next section, the overview of the filter process is outlined. The two constituent processes of the filter process are explained and their working is illustrated.

## 5.3. Filter process

As stated in previous chapters, the purpose of the filter process in the pre-processing stage is to group the essay dataset according to the quality of the essays, so that appropriate techniques can be used to grade them further. The complete filter process is illustrated in figure 5.2 below.

This process is sub-divided into two stages and is diagrammatically represented in Figure 5.2. The two stages in the filter process are filter process stage 1 (FPS 1) and filter process stage 2 (FPS 2).

### 5.3.1. Filter Process Stage 1 (FPS 1)

In FPS 1, anomalous essays are detected by the system. An 'anomalous' essay can be defined as an essay that has an undesired illustration or characteristic. There are five different types of anomalous essays and for each type an action needs to be undertaken to filter it out. The different types/cases of anomalous essays and the required action taken by the filter process in each case are listed in Table 5.1.

In order to identify if the essay is an anomalous essay case 1 (blank response), the action is to scan the essay for null characters. If the essay is a blank response, then it is stored in 'Anomalous essays' dataset. In order to identify if the essay is an anomalous essay case 2 (a picture) the action is to scan the student response for drawing. If the essay is found to be a picture, then it is stored in 'Anomalous essays' dataset. In order to identify if the essay is an anomalous essay case 3 (a picture and words) the action is to scan the student response for drawing and characters. If the essay has a picture and words then the words are to be graded for 'spelling' according to the NAPLAN rubric. In order to identify if the essay is an

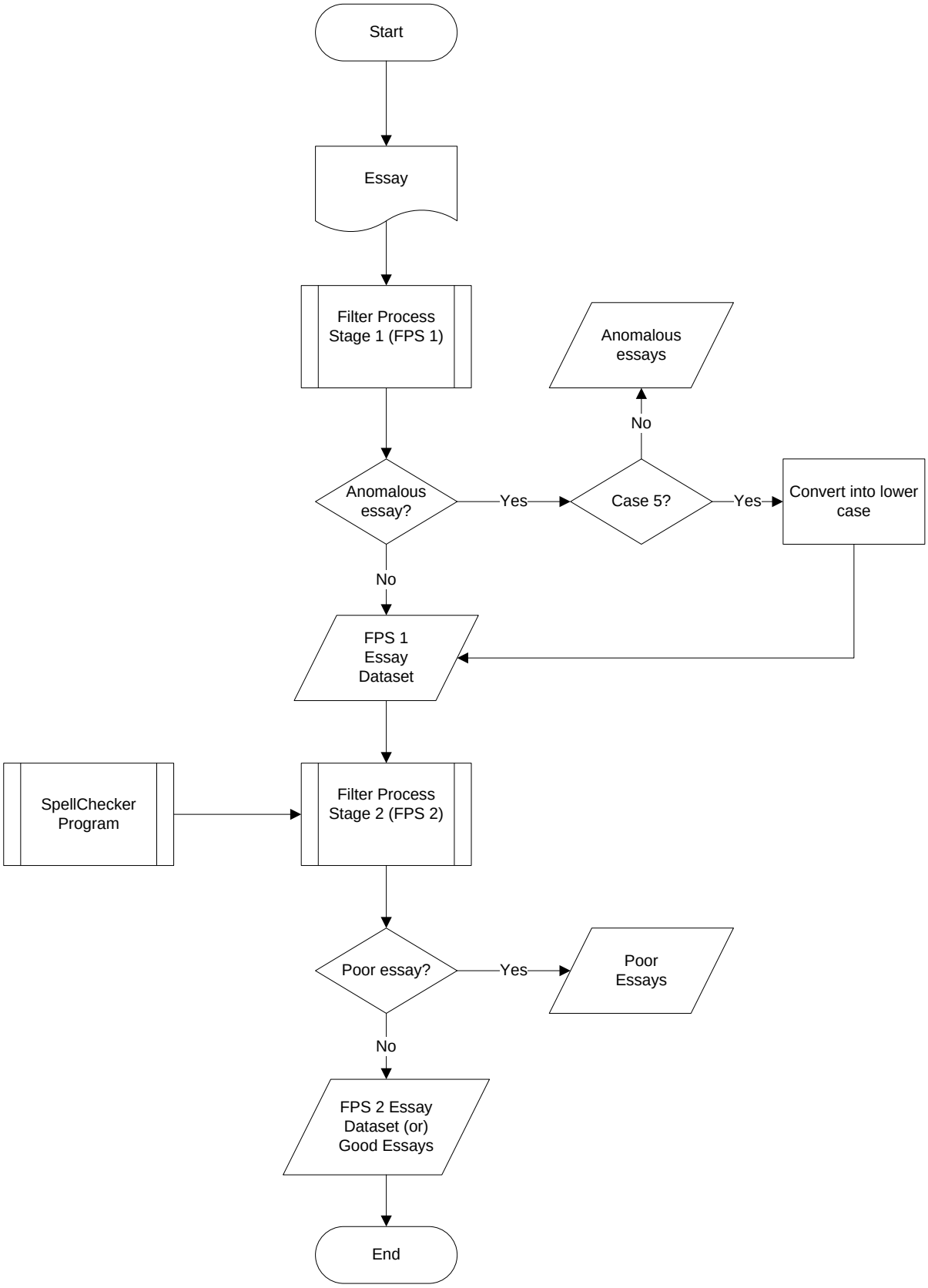



**Figure 5.2: Overview of the Filter Process**

**Table 5.1: Anomalous Essays Cases 1 to 5**

| Case | Type of Anomalous essay | Required Action to detect the essay |
|---|---|---|
| 1 | Blank response (student submits a blank paper) | Scan the essay for null characters and store essay in 'Anomalous Essays' |
| 2 | Drawing a picture (student draws a picture instead of writing an essay) | Scan the essay for drawings and store essay in 'Anomalous Essays' |
| 3 | Drawing a picture and writing some words | Scan the essay for drawings and grade the words for 'Spelling' |
| 4 | Copying the question prompt as answer - partly or completely | Find similarity between question prompt and answer essay |
| 5 | Writing essay completely in upper-case letters | Convert to lower-case letters and send the essay to 'FPS1 Essay Dataset'. |

anomalous essay case 3 (a picture and words) the action is to scan the student response for drawing and characters. If the essay has a picture and words then the words are to be graded for ‘spelling’ according to the NAPLAN rubric. In order to identify if the essay is an anomalous essay case 4 (copied question prompt as answer) the action is to find the similarity between question prompt and answer essay. If the similarity is more than 30% (empirically derived value) then the essay is stored in ‘Anomalous essay’ dataset else it is sent to the ‘FPS1 essay dataset’. If the essay is an anomalous essay case 5 (essay written completely in upper-case letters) then the action is to convert the essay to lower-case letters and send the essay to the output of this stage ‘FPS1 Essay Dataset’.

The essays in the ‘anomalous essays’ dataset are assigned a score of ‘0’. The FPS1 essay dataset is the input to the next stage of the filter process. This stage is called the Filter Process Stage 2 or the FPS2. At this stage, the FPS 1 Essay Dataset goes through more filtration in order to separate 'noisy' essays called "poor essays" from the dataset. The FPS2 is explained in detail in the next sub-section.

### 5.3.2. Filter Process Stage 2 (FPS 2)

In order to detect poor essays in the FPS1 essay dataset, it is required to extract some surface feature values of the essays. For this purpose, FPS 2 comprises two steps. In step 1, automated feature extraction is performed and in step 2, poor essays are detected. The automated feature extraction is performed by scanning the FPS 1 Essay Dataset, one essay at a time, through the Spellchecker program. It is a custom-written program in C# and it is used to extract surface features from the essay. For the purpose of FPS2, we need to obtain the number of spelling errors, the number of grammar errors (an error violating one or more rules of English grammar is called a grammar error [114]), the total number of paragraphs and the total number of words in each essay. The automated feature extraction is explained in further detail in the next sub-section.

#### 5.3.2.1. Automated Feature Extraction

The input essay dataset for the filter process consists of essays as Word documents. Microsoft Word is a word processor that can be used to compose, edit and format any type of writing, either formal or informal. It can be used to write letters using informal language or even scientific research publications, where the language needs to be more formal. It has a range of settings that can be customised to suit our requirements. Since we are dealing with narrative essays for the purpose of this thesis, we choose the following settings in Word 2007.

1. In language settings, we select the “Primary Editing Language” as “English (Australia)”.

2. In proofing settings, we ignore internet addresses, file addresses and repeated words.
3. In spelling and grammar correction, we choose contextual spelling, display readability statistics and check grammar with spelling.
4. In grammar settings, we choose all options including punctuation and sentence structure.

A customised program “SpellChecker” is written in C# language and each Word document is processed through it to extract several surface features from the essay. Figure 5.3 shows the screenshot of the SpellChecker program.

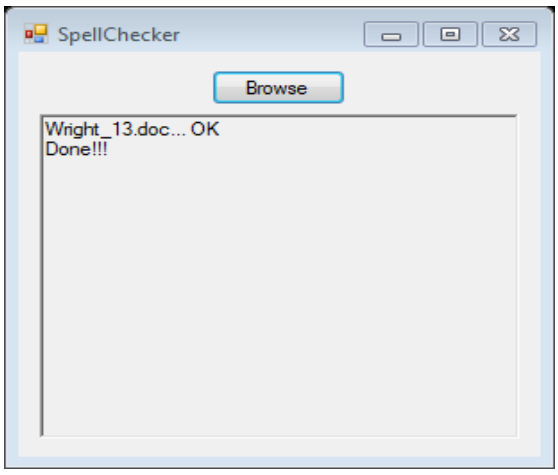



**Figure 5.3: Screenshot of ‘SpellChecker’ program**

In figure 5.3, the essay to be processed can be selected by using the “Browse” button. Then the essay name is displayed (Wright_13.doc in the screenshot). In a few seconds, the essay is processed and the message “OK… Done!!!” is displayed by the program. The result for the essay is displayed by the program in XML format, as shown in figure 5.4.

The various features extracted by the SpellChecker program, as shown in figure 5.4, are:

1. Total number of spelling errors, a list of words that have a spelling error and the first suggestion for every spelling error.
2. Total number of grammar errors and the list of sentences that have a grammar error.

3. Number of pages, number of words, number of characters (with space), number of characters (without space), number of paragraphs, number of lines. These are obtained from the 'Word Count' option in Word.
4. Values of number of sentences, number of sentences per paragraph, average sentence length, average word length, percentage of passive sentences, Flesch Reading ease value and Flesch-Kincaid Grade level value. These are obtained from the 'Readability Statistics' in Word.

The output from the SpellChecker is in XML format, as shown in figure 5.4. From the output, we use only four features for the filter process and use the rest in other modules for grading vocabulary and sentence structure which are discussed later. The four features used in the filter process are number of spelling errors, number of grammar errors, number of paragraphs and total number of words in the essay as shown in next sub-section. Complete results of SpellChecker program can be found in Appendix C.

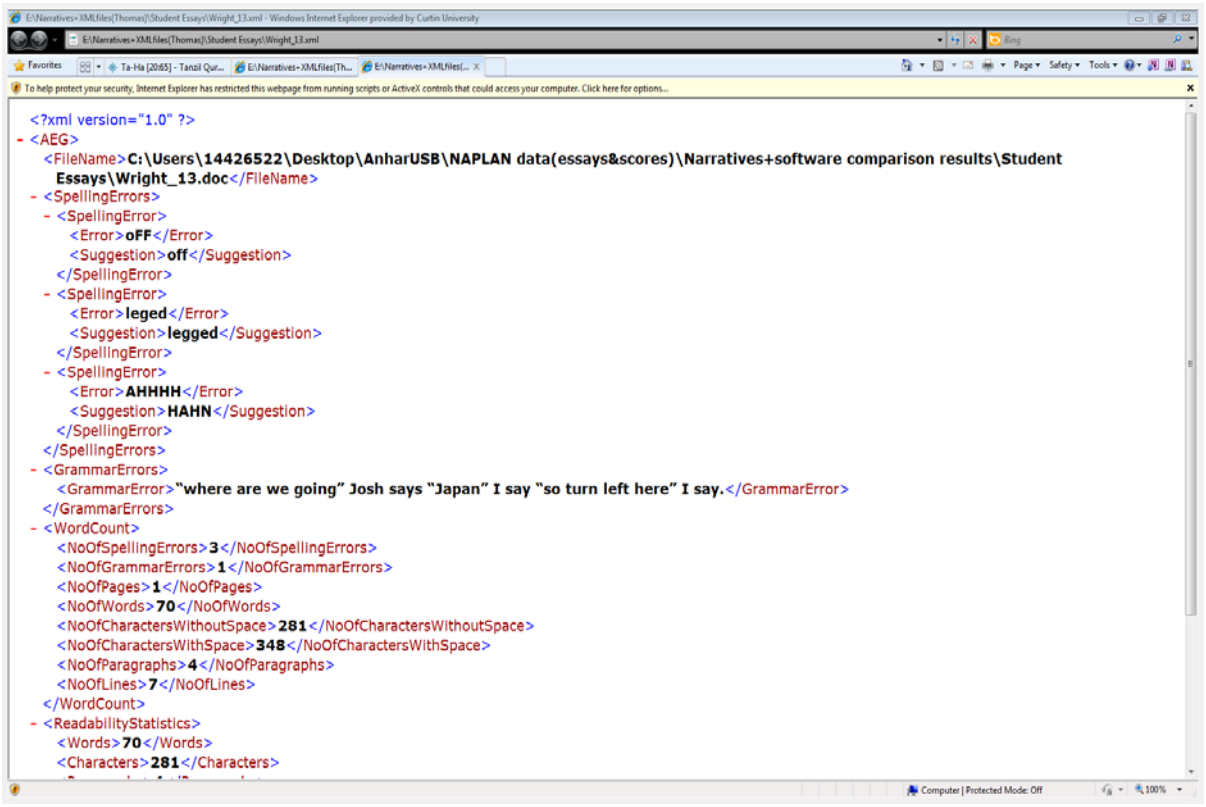



**Figure 5.4: Output from 'SpellChecker' program.**

#### 5.3.2.2. Detection of Poor essays

The mathematical formulation for detection of poor essays is shown in figure 5.5 below. There are two cases of poor essays. At least one of them needs to be satisfied in order for an essay to be classified as a 'poor' essay. Case 1 is when essay is extremely poor in spelling and punctuation. For case 1, the normalised spelling error (NSE) and the normalised grammar error (NGE) for an essay are calculated as shown in figure 5.5. NSE is calculated by dividing the total number of spelling errors (SE) by the total number of words (N). If NSE $\geq$ 0.1, then case 1 is true. NGE is calculated by dividing the total number of grammar errors (GE) by the total number of sentences in the essay (S). We consider S because Microsoft Word provides grammar errors on a sentence level, in contrast to spelling errors which are provided on a word level. As shown in figure 5.5, if NGE $\geq$ 1, then case 1 is true. The values of 0.1 for NSE and 1 for NGE are obtained empirically.

Case 2 is when essay length is less than or equal to 80 words. For case 2, we consider the value of total number of words in the essay. According to the Standard American English Dictionary, an 'essay' is "a written answer that includes information and discussion, usually to test how well the student understands the subject". For instance, in a narrative essay, the writer tells a story or part of a story. Also, a narrative essay is comprised of an orientation, complication and a resolution [8]. Hence, a minimum of 80 words is desired in an essay.

If (NSE $\geq$ 0.1) OR (NGE $\geq$ 1) then instance/essay = Poor Essay where, NSE = SE/W and NGE = GE/S

Else If W $\leq$ 80 then instance/essay = Poor Essay

**Figure 5.5: Mathematical formulation for detection of poor essays.**

The essays that satisfy either case 1 or 2 are detected as poor essays and are stored in the "Poor Essays" dataset. All the remaining essays of this phase i.e., the essays that are neither anomalous nor poor, are stored in the "FPS 2 Essay dataset" and are treated as good essays.

The proper division of the essay dataset into three sets is of paramount importance because in the grading stage, separate algorithms are used to grade poor essays and good essays. The three separate datasets are the actual input to the "Grading process" where three different modules incorporate appropriate grading logic for each dataset. The complete grading process is explained in detail in later chapters.

In the next section, the testing undertaken for the filter process and the results obtained are explained.

#### 5.3.2.3. Testing and Results

The test essay dataset consists of 308 student essays provided to us by WA-DET. All these essays were first annotated by a human marker into one of the three categories: anomalous, poor and good essay. Then the dataset was run, one essay at a time, through the filter process. Table 5.2 gives the number of noisy essays detected in each case/category of anomalous and poor essays.

**Table 5.2: Number of 'noisy' essays detected (out of 308 essays)**

| | | |
|---|---|---|
| Anomalous Essays | Case 1-Blank response | 2 essays |
| | Case 2-Drawing a picture | 1 essay |
| | Case 3-Drawing a picture and writing some words | 3 essays |
| | Case 4-Copying question prompt as answer | 3 essays |
| | Case 5-Writing completely in upper-case letters | 2 essays |
| | Case 1- Extremely poor in spelling and punctuation | 65 essays |

| | | |
|---|---|---|
| Poor Essays | Case 2- Essay length ≤ 80 words | 70 essays |

Out of a dataset of 308 essays, a total of 11 anomalous essays and 135 poor essays were detected successfully, as shown in table 5.3. The number of anomalous essays detected in cases 1, 2, 3, 4 and 5 were 2, 1, 3, 3 and 2 respectively. On the other hand, the number of poor essays detected in cases 1 and 2 were 65 and 70 respectively. The results of the testing are discussed in table 5.3.

The actual number of anomalous essays was 13 but only 11 essays were correctly detected during the filter process. The remaining two essays were not detected. On the other hand, there were 129 poor essays in the dataset but 135 were detected as poor essays. Furthermore, there were 168 good essays but 172 were detected as good essays. Looking at table 5.3, it might seem that the performance of the filter process is quite good because the agreement between the actual number of essays determined by the human annotator and the detected number of essays in the output obtained from the filter process is relatively high (the maximum difference between them is 7).

**Table 5.3: Results obtained from the filter process.**

| Essay Category | Actual number | Detected number |
|---|---|---|
| Anomalous essays | 13 | 11 |
| Poor essays | 129 | 135 |
| Good essays | 168 | 172 |

However, to accurately determine the performance of the process, we need to calculate precision, recall and F-measure. These performance evaluation metrics are commonly used in information extraction and pattern recognition techniques, to determine the accuracy of a classi-

fication model. Since we are basically classifying an essay into one of the three classes, we can use these metrics to determine the accuracy of the filter process.

#### 5.3.2.4. Performance Evaluation

The formulae for the computation of performance metrics precision, recall and F-measure are as follows:

Precision = $\frac{TP}{TP+FP}$ ……Equation 5.1

Recall = $\frac{TP}{TP+FN}$ ……Equation 5.2

F-measure = $2 * (\frac{Precision*Recall}{Precision+Recall})$ ……Equation 5.3

Where:

- TP = True Positives, the number of essays correctly classified into a category.
- FP = False Positives, the number of essays incorrectly classified into a category
- FN = False Negative, the number of essays incorrectly classified into other categories

Ideally, the value of these metrics should be as close to 1 as possible. From the output of the filter process, the value of TP, FP and FN is calculated for each essay category and then these metrics are calculated manually.

The precision, recall and F-measure values for each category are shown in table 5.4. Overall, the performance values vary little between each category. The average values for precision, recall and F-measure are computed and these values indicate the performance of the filter process. Hence, the Filter process performs with a precision, recall and F-measure of 0.92, 0.96 and 0.97 respectively, which is very promising. Now the next step is to analyse the results of the testing process and delve deeper into the reasons for the errors encountered in this process. This is further discussed in the next section.

**Table 5.4: Performance evaluation of Filter Process**

| Essay Category | TP | FP | FN | Precision | Recall | F-measure |
|---|---|---|---|---|---|---|
| Anomalous essays | 11 | 0 | 2 | 1.00 | 0.96 | 0.97 |
| Poor essays | 129 | 6 | 5 | 0.85 | 0.96 | 0.98 |
| Good essays | 168 | 5 | 4 | 0.92 | 0.96 | 0.97 |
| Average value | | | | 0.92 | 0.96 | 0.97 |

#### 5.3.2.5. Discussion and future work

From table 5.4, it is evident that the FN values for a total of 12 essays were placed in the wrong category by the filter process. The two kinds of essays that escaped detection during the filter process are: (i) essays where the student used both upper case and lower case letters in an alternating manner, as in "ThEre WaS A dOg". In order to detect this, the condition for anomalous essays, case 5 needs to be adjusted accordingly; and (ii) essays that are written in an unconventional way, as an interview or as diary entries. Although these kinds of essays are rare, we still need to detect them in order to grade them properly. Some essays that were actually good essays were detected as poor because they satisfied case 1 of poor essays. This shortcoming can be resolved as follows.

Despite the fact that the noise reduction methodology has shown very good performance so far, there is still scope for improvement. We acknowledge that the condition in case 1 for 'Poor Essays' is quite high (0.1). It is desirable that the threshold be decreased further to 0.2 or 0.3. Otherwise, there is a problem of having too many poor essays being filtered out. This would happen mostly because students in primary and secondary school are still learning and experimenting with language and are thus bound to make many spelling and punctuation er-

rors. In future, it is recommended that a filter of the filter process be created in order to address this problem.

Overall, the performance and results produced by the filter process satisfy our requirements very well. After obtaining the three separate datasets for anomalous essays, poor essays and good essays, appropriate grading techniques are applied to each set in the grading process. A neural network model performs the grading in order to grade the essays for various criteria. In the next section, the feasibility analysis of the neural network model is presented.

## 5.4. Feasibility analysis of the neural network model

As discussed earlier in chapter 4, section 4.2.2.2, in order to model both the linear and the non-linear relationship between the feature vector and the essay grade, we propose to use neural networks. Neural Networks have been widely used for linear and non-linear function fitting, function approximation, time series prediction, pattern classification and sequence identification in various domains such as text mining, robotics, power systems [112]. However, in the domain of automated essay grading, only two existing systems used it so far. But the many benefits of these networks for the purpose of AEG are yet to be uncovered.

The term 'multivariate nonlinear regression' refers to nonlinear regression with two or more predictors ($x_1$, $x_2$,…, $x_n$). When multiple predictors are used, the nonlinear relationship cannot be visualized in two-dimensional space [119]. Hence, the non-linear relationship between the various features of an essay and the essay grade cannot be visualized in two-dimensional

space. To perform feasibility analysis on using a neural network model for essay grading, we first need to design a neural network model. We design a multi-layer feed forward neural network with back propagation algorithm as the training algorithm. Using the MATLAB GUI tool for Neural Networks - 'nntool', we created Multi-Layer Feed Forward Neural (MLFFN) networks of various configurations. For each network, training was performed using the training set. The network was calibrated by retraining over a number of iterations until it produced a low mean squared error (MSE) value and the network performed satisfactorily. Then using the testing set, the network was simulated and the results are reported. The next step was to perform null hypothesis testing in order to ascertain that there is no significant difference between the targets and the results obtained as suggested in [120], by performing a Student's t-test [121]. If the null hypothesis testing is successful, then our system will be deemed as feasible.

### 5.4.1 Experimental Simulation and Testing

Using the validation technique of 'stratification', the essay dataset was divided manually into two sets: a training set and a testing set. According to this technique, the training set comprises of 80% of data and is used for the training and calibration of the network. The remaining 20% of data is called testing set and is used to test the performance of the network. It is important to note that the network has not seen the testing set before. Hence, it is used only after completing the training phase of the network. This is also called the 'Hold-Out method' in data mining [120]. Our dataset consists of 172 good essays. Hence, using the above technique, the training set consists of 138 essays and the testing set consists of 34 essays.

When the training set is provided to the neural network, the network splits it into three subsets: the training subset, validation subset and testing subset. This is done randomly and automatically, the data is split in the ratio of 70%, 15% and 15% to obtain the three subsets. Ac-

cordingly, the training set of 138 essays was split into a training subset consisting of 84 essays, a validation subset and a testing subset each consisting of 27 essays.

**Table 5.5: Neural network architecture for feasibility analysis**

| | |
|---|---|
| Type of neural network | Multi-Layer Feed Forward Neural Network (MLFFN) |
| Number of hidden layers | 2 |
| Number of neurons in hidden layer | 45, 50, 55 |
| Training algorithm | 'trainlm' |
| Learning Algorithm | Back propagation |
| Learning functions | 'Tansig', 'purelin' |

As shown in table 5.5, we configured three different MLFFN networks with 2 layers in the hidden layer for each network. The first network had 45 neurons in the hidden layer. Then we increased the number of neurons by 5, so the second network has 50 neurons in the hidden layer. We performed the training and testing of this network in the same way as for the first network. Then we increased the number of neurons in the hidden layer by 5. Hence, the third network has 55 neurons. For each network, the training algorithm is 'trainlm' and the learning algorithm is 'back propagation'. The learning functions for hidden layer 1 and hidden layer 2 are 'tansig' and 'purelin' respectively because this is a function fitting task. Validation is performed by increasing number of neurons in hidden layer and repeatedly checking the performance of the network until the performance decreases after a certain amount of neurons in the hidden layer.

Using the GUI shown in figure 5.6, the data for the training phase is chosen. The inputs and targets are specified and then the network is trained. The outputs of this phase are stored as training results.

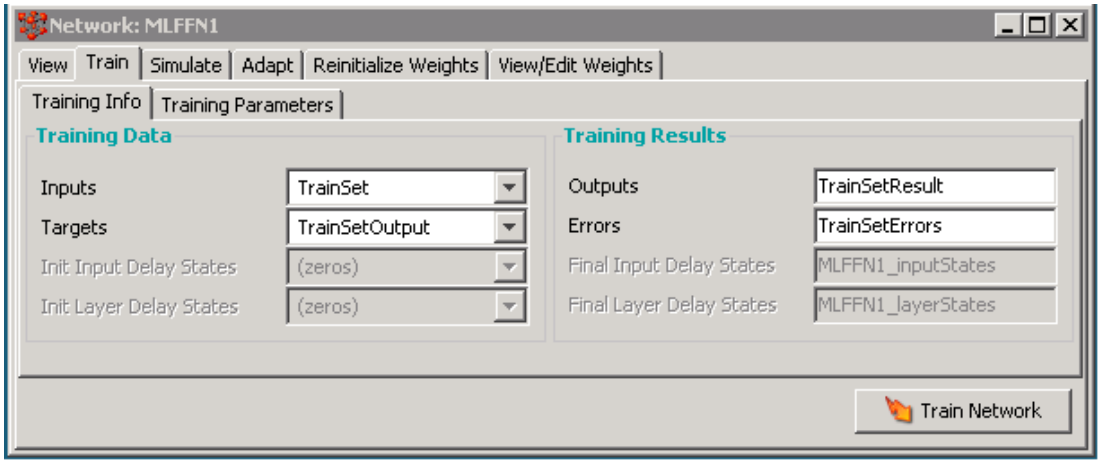



**Figure 5.6: MATLAB GUI to design network**

Figure 5.7 shows the GUI of the MATLAB neural network training tool called 'nntraintool'. It shows the design of the neural network consisting of the input layer which is responsible for feeding the inputs to the hidden layer. In the hidden layer, the learning function is 'tansig', the training algorithm is 'trainlm', the performance function for the network is 'mse', and the data division is set at random, as mentioned earlier. In the next hidden layer, the result obtained from the network is converted back to the format of the expected results by using a simple 'purelin' function as the learning function. This output is sent to the output layer as the result obtained from the neural network.

For training the network, all the default settings of training parameters are used. Training stops when validation checks are performed six times. Furthermore, the GUI also shows various plots that help us determine the performance of the network. The performance plot 'plotperform' shown in figure 5.8 illustrates the best validation performance of the network. There

are three curves shown in the plot. It can be seen in the plot that MSE during all three phases of training, validation and testing, was initially very high but reduces rapidly. The best value

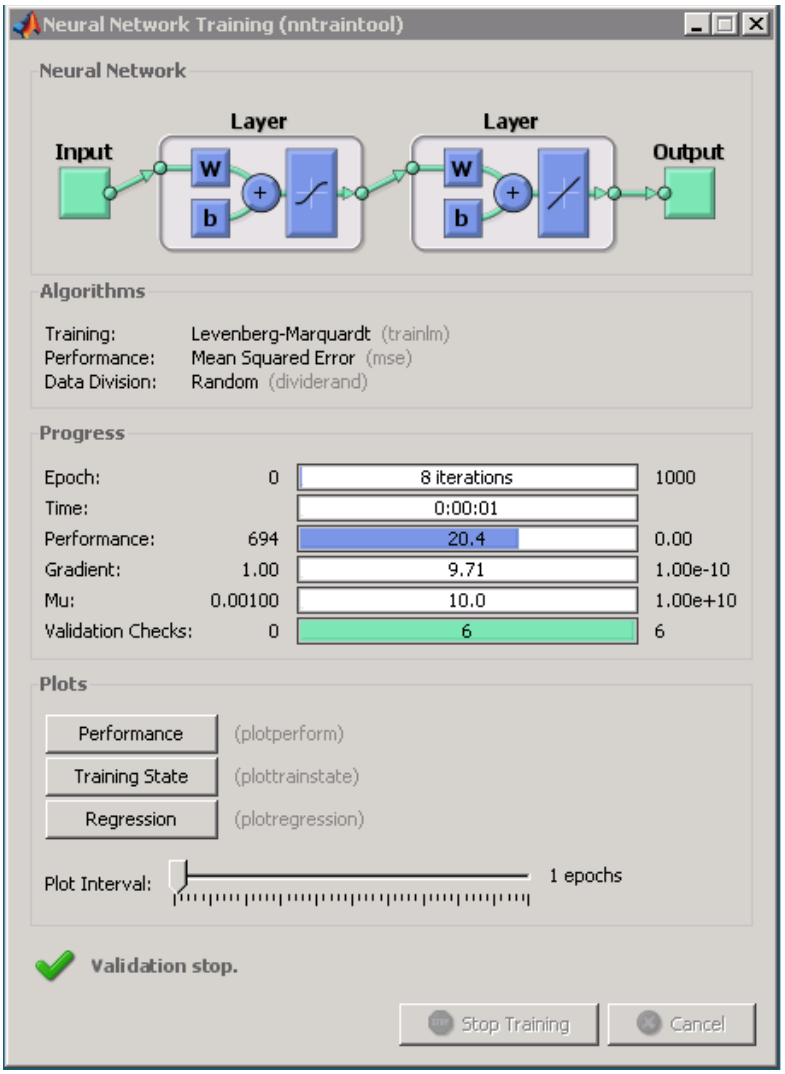



**Figure 5.7: Screenshot showing the network details.**

of MSE during training was reported as 33.18. If the MSE values during validation and training also follow a similar pattern to the training phase, then training stops.

The results obtained during the training phase of each network were recorded and used to calculate the root mean square error (RMSE) using the formula 5.4 below.

RMSE = $\sqrt{\frac{\sum (f(x_i) - y_i)^2}{n}}$ ……..…Equation 5.4

where $f(x_i)$ = Target value of $i^{th}$ essay,

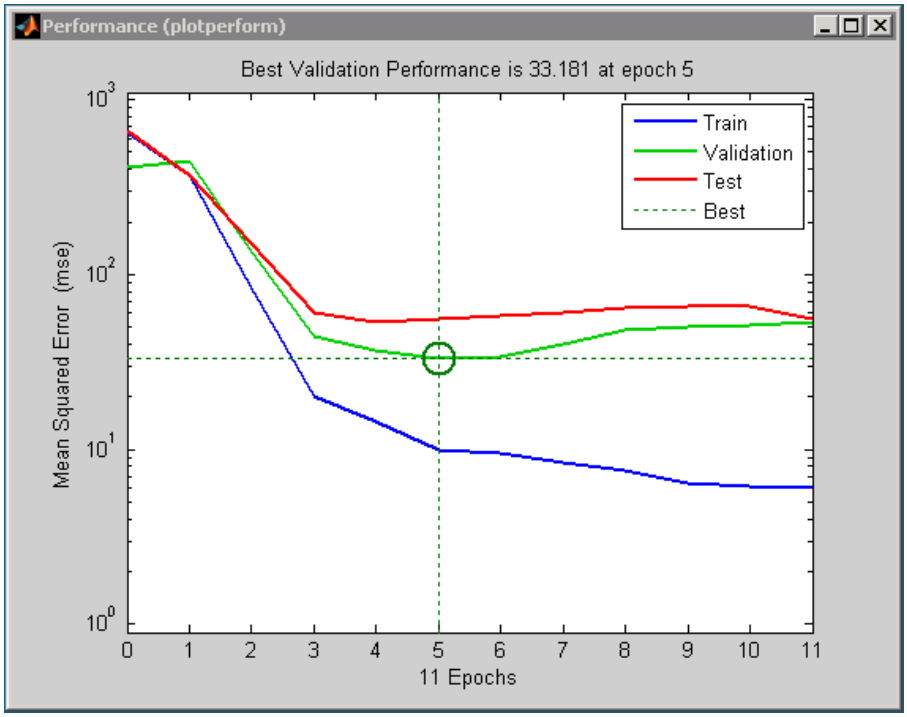



**Figure 5.8: Screenshot showing mse in each phase of network**

$y_i$ = Result obtained for $i^{th}$ essay,

n = Total number of essays in dataset.

In figure 5.9, the RMSE values for networks with various configurations are shown. For networks with N=45, N=50, N=55, the RMSE values are 5.27, 5.81 and 6.38 respectively. The network with neurons = 45, is chosen as the final model because it produced the lowest mse.

The network with neurons = 45, is chosen as the final model because it produced the lowest mse. The network with the optimal performance is the one which provided the least MSE and hence, it was chosen as the final model.

Then, this network is simulated using the testing set. Using the GUI shown in figure 5.10, the data for the testing phase is chosen. The inputs and targets are specified and then the network is simulated. The outputs of this phase are stored as simulation results.

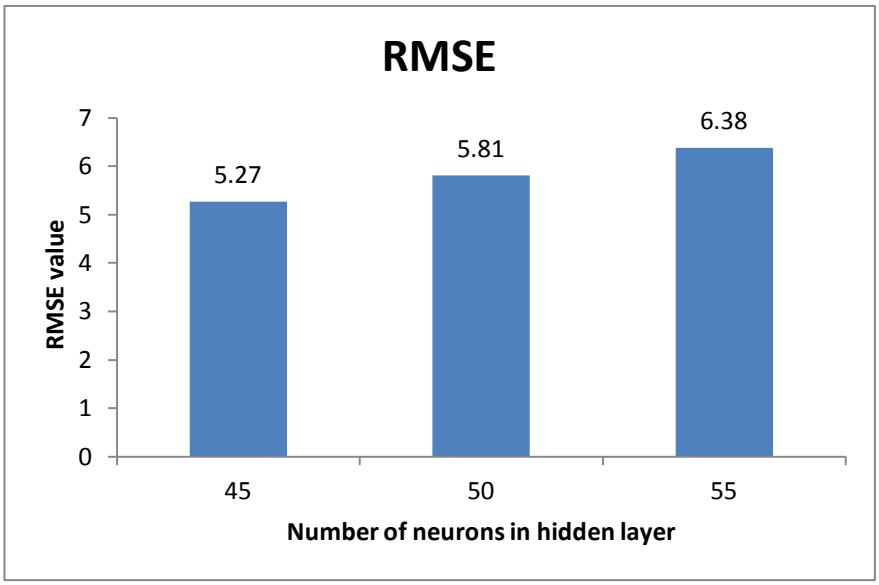



**Figure 5.9: RMSE values in the training phase.**

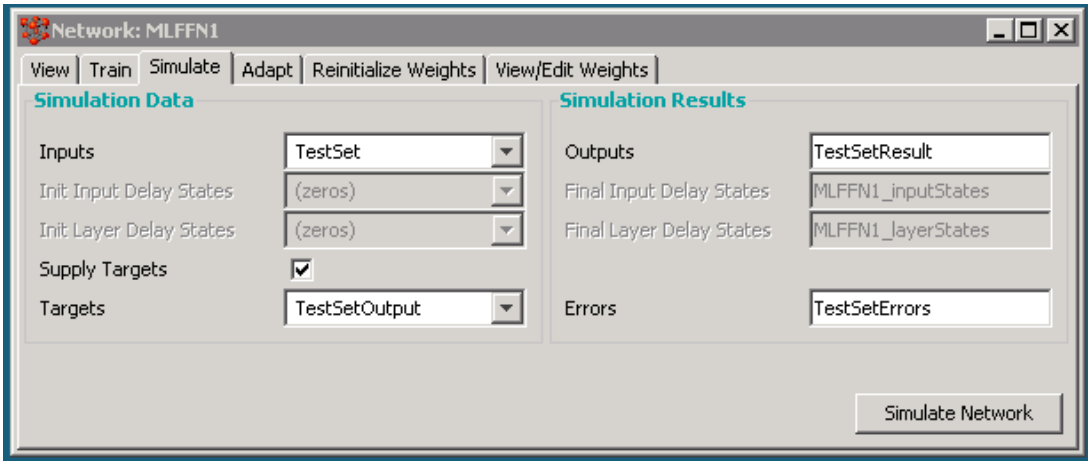



**Figure 5.10: Screenshot showing the testing phase details.**

### 5.4.2. Performance Evaluation

In order to evaluate the performance of the system and to analyse the feasibility:

1. the results obtained from the final network model and the targets (actual scores of essays) are used.
2. The mean and its 95% confidence interval values are calculated for results obtained and for the targets, using the formula below. The values obtained are shown in figure 5.11. Since the two confidence intervals overlap in the figure, this means that at a 95% level of confidence, there is insufficient evidence that the two means are different, according to [5].

95% Class Interval limits = $x \pm z^*(S.D/\sqrt{n})$…….Equation 5.5

Where x = Absolute mean value of sample

z = 1.96

S.D = Standard Deviation of sample

n = Size of sample

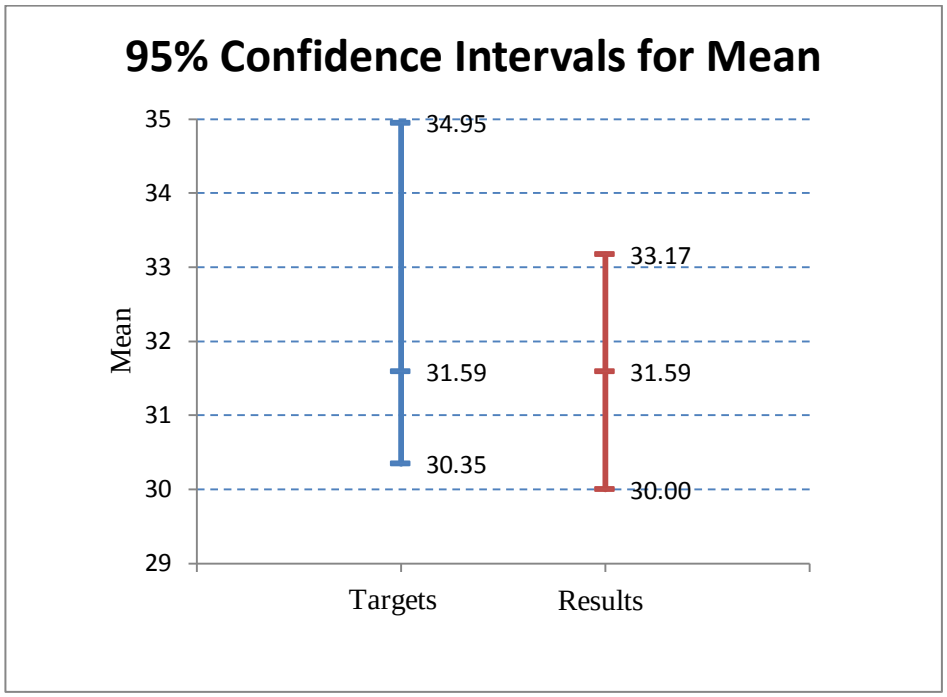



**Figure 5.11: Values of mean obtained from targets and results.**

Furthermore, to corroborate the above point, we need to test the hypothesis that there is no significant difference between the targets and the outputs as suggested in [5], by performing a Student's t-test [6]. We have formulated the null hypothesis $H_o$ as shown below to denote that the difference between the targets and the outputs is statistically insignificant.

Null hypothesis testing: $H_o \rightarrow \mu_1 = \mu_2$

where $\mu_1$ = Targets and $\mu_2$ = Results obtained from the neural network model.

Then, we performed the t-test and determined the critical p values at significance level, $\alpha$=0.05 and degrees of freedom (dF) = N – 1 = 34 – 1 = 33. The t-test was paired because of dependent samples and 2-tailed because of unequal variances. The formula used to calculate t value is given below.

$$t = \frac{\overline{X}_D - \mu_0}{s_D/\sqrt{n}}. \quad \text{.....Equation 5.6}$$

Where $X_D$ bar = absolute average difference between the two samples

$\mu_0$ = non-zero constant

$S_D$ = standard deviation of difference between the two samples

n = size of sample

The results obtained in the t-test are t = 0.27754, $p_{(\alpha = 0.05)}$ = ± 1.59 and dF = 33.

Since the t value lies inside the interval of the p value, we accept the null hypothesis that there is no significant difference between the targets and results. In other words, the difference between the actual essay scores and the scores assigned by the neural network is statistically insignificant.

We can thus conclude that our feasibility study has ended favourably. Furthermore, it indicates that it is possible to assign essay scores by using a neural network.

In the next section, feature optimization is undertaken to determine which of the four features influence the essay grade.

### 5.4.3. Feature Optimization

The neural network model described in the section above was used to determine the order of influence of the features on the essay grade. This is also called 'feature optimization' in the field of data mining. To achieve this, we implemented the widely used 'leave-one-out' method to find influential features, as reported in [7]. According to this technique, one feature/criterion is left-out from the four input criteria while we fed the remaining three criteria into the MLP FFN. Firstly, the feature 'Total paragraphs' in the essay was left out and the model was provided inputs with only values of grammar errors, total number of words and spelling errors. All these values were extracted from the 172 good essays using the Spell-Checker program, which was explained previously in this chapter. With the three inputs, the neural network system was trained until the network performed satisfactorily by providing a low mse. Afterwards, the network was simulated with the testing set and results were obtained. Then the RMSE values were calculated from the results obtained from the network. The value of RMSE when each feature was left out is given in figure 5.12. The value of RMSE when 'Total paragraphs', 'Grammar errors' ,'Total Words' and 'Spelling errors' were left out was 5.42, 6.41, 26.11 and 6.28 respectively as shown in figure below.

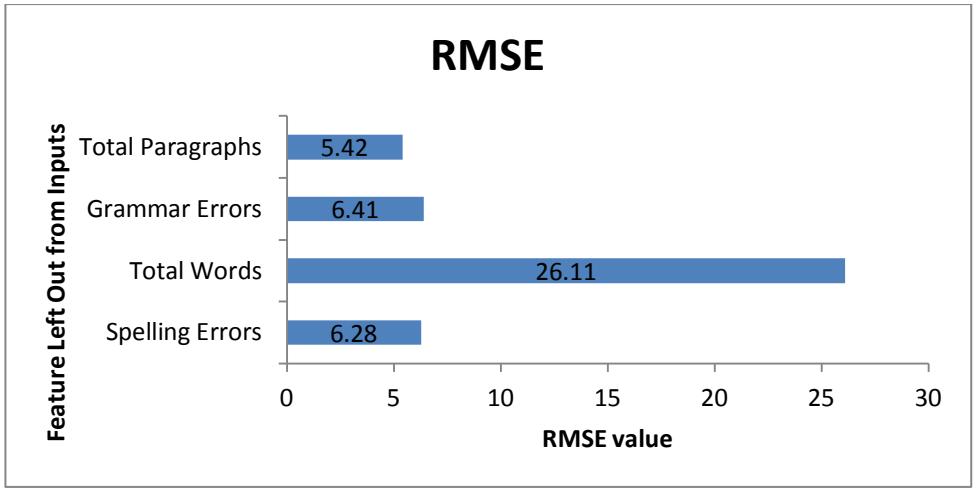



**Figure 5.12: RMSE values for feature optimization.**

The feature whose absence produced the highest mse means that it is most influential on the grade. Accordingly, from figure 5.12, it can be concluded that our results showed that the best predictor of an essay score is the 'number of words in essay'. This finding is in line with [8], where the total number of words in an essay is proved to be a very strong predictor of an essay score.

Then, in order of highest to lowest influence on the essay grade, the features are grammar errors, spelling errors and number of paragraphs.

## 5.5. Conclusion

In this chapter, an overview of the filter process and its sub-stages, filter process stage 1 and filter process stage 2, was described in detail. The testing of the filter process was described and an evaluation of the working of the filter process was given, showing that the filter pro-

cess performed very well. Then the feasibility analysis of the neural network model was provided. It is important to conduct a feasibility analysis so that we can prove that our grading system will perform in line with the grading done by human markers. Furthermore, a favourable feasibility analysis of the grading system will also enable us to demonstrate that when it is developed for the modules of vocabulary and sentence structure, it will work optimally. We showed the various stages in the experimental simulation and performed an evaluation of the results obtained.

In the next chapter, we explain in detail the methodology of the automated grading of the spelling module.

## 5.6. References


[1] D. Naber, "A Rule Based Style and Grammar Checker," Masters thesis, Faculty of Computer Science, University of Bielefeld, 2003.

[2] Department of Education, *Narrative Marking Guide 2010, National Assessment Program-Literacy and Numeracy*: Government of Western Australia, 2010.

[3] T. S. Dillon and D. Niebur, *Neural networks applications in power systems*: CRL Publishing Ltd, Market Harborough, UK, 1996.

[4] Oracle®. (2008). *Data Mining Concepts. 11.1*. Available: http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28129/regress.htm

[5] A. Flexer, "Statistical evaluation of neural network experiments: Minimum requirements and current practice," in *Proceedings of the Thirteenth European Meeting on Cybernetics and Systems Research*, 1996, pp. 1005-1008.

[6] M. R.Spiegel, *Schaum's Outline Series Theory and Problems of Statistics. 2/ed.* London: McGraw Hill, 1992.

[7] Y. Marchand, C. R. Adsett, and R. I. Damper, "Evaluating Automatic Syllabification Algorithms for English," presented at the 6th ISCA Workshop on Speech Synthesis, Bonn, Germany, 2007.

[8] Y.-W. Lee, C. Gentile, and R. Kantor, "Toward Automated Multi-trait Scoring of Essays: Investigating Links among Holistic, Analytic, and Text Feature Scores," *Applied Linguistics*, vol. 31, pp. 391-417, 2010.

# Chapter 6: Automated Marking of Spelling

## 6.1. Introduction

Previously in chapter 4, we described the conceptual framework of the spelling module. In this chapter, the methodology of the spelling module is presented in detail. This module is responsible for grading the spelling in an essay and assigning the appropriate score from a band of 0-6, according to the NAPLAN rubric. In order to grade spelling for an essay, the first step is to identify the correctly spelled words and the incorrectly spelled words, for which we use our custom developed program "SpellChecker". The second step is to classify each correct and incorrect word into one of the four classes – simple, common, difficult and challenging, for which we use the Word Classification algorithm. The third and final step is to assign the score for spelling, for which we use the Spelling Mark algorithm. In this chapter, each step is described in detail, after which the working of the algorithms is explained with the help of examples. Finally, the testing and evaluation of the algorithms is provided and a discussion of the results is presented.

## 6.2. Automated Scoring of Spelling

Figure 6.1 shows the complete model for the automated scoring of spelling. The various steps involved in the automated scoring of spelling are as follows.

1. The first step is to obtain the total number of words in an essay and the number of spelling errors in that essay. We obtain these values from the XML output provided by the "SpellChecker" program. If the number of words is same as the number of spelling errors, then the spelling score is assigned as '0'. In other cases, for an essay, we obtain the first suggestion for every spelling error from the output of the "SpellChecker" program. All the words which do not appear in the spelling errors are treated as correctly spelled words.
2. The next step is to create two lists: one list containing correctly spelled words in the essay and another list containing the suggestions for spelling errors. These lists are populated with the condition that words are not repeated in the list and each word is mentioned only once. In this way, we consider each word only once for spelling, irrespective of the number of times it is written in the essay.
3. The next step is to find the class of each word in both the lists. To do this, we need an automated program for the classification of words. Section 6.3 describes the Word Classification program further.
4. After obtaining the class of every word, the next step is to calculate the percentage of 'simple' words and 'common' class.
5. The next step is to assign the score for spelling, for which we need an automated program. Section 6.4 describes the Spelling Mark Algorithm further.

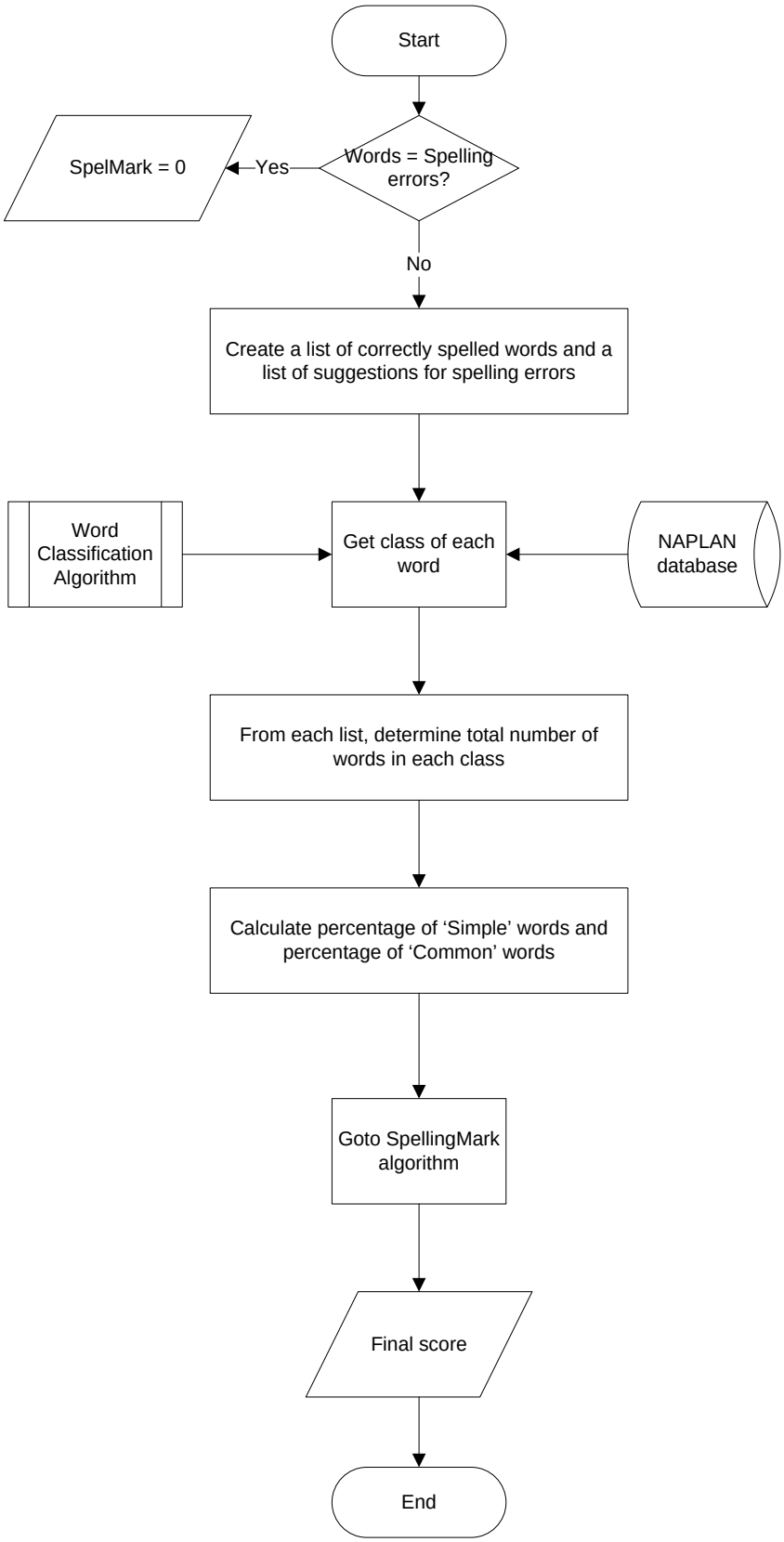



**Figure 6.1: Methodology for automated marking of spelling**

## 6.3. Word Classification Algorithm

The word classification algorithm is based on certain heuristics and rules based on the English language. Conventional English language concepts are used to formalise the NAPLAN guidelines for word classification. To serve as a foundation for understanding this algorithm, we defined the various terms involved, in Chapter 4, section 4.2.2.1.1. Further to this and for the purpose of developing this algorithm, we divide consonant di-blends into two sets, based on their position in a word. One set of consonant di-blends can occur anywhere in a word, as mentioned in point 4 in Table 6.1 below. Another set of consonant di-blends can occur only in the ending of a word, as given in point 5 in the same table. Consonant tri-blends can occur anywhere in the word. Consonant di-graphs can also occur anywhere in the word [1].

For the purpose of this algorithm, we state the following points in table 6.1 below.

**Table 6.1: Terms used in Word Classification Algorithm and their definitions**

| Point | Condition |
|---|---|
| 1 | Alphabet = [b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z] |
| 2 | Vowel = {a, e, i, o, u} |
| 3 | Consonants (C) = Alphabet – Vowel |
| 4 | Consonant di-blends Type 1 (anywhere in word) = {br, cr, dr, fr, gr, pr, tr, sc, sk, sl, sm, sn, sp, st, sw, bl, cl, fl, gl, pl}. |
| 5 | Consonant di-blends Type 2 (ending in word) = {ct, ft, ld, lp, lt, lk, mp, nd, nk, |

| | |
|---|---|
| | nt, pt, rd, rk, sk, sp, st}. |
| 6 | Consonant tri-blends = {str, spr, thr, chr, phr, shr, nth, sch, scr, sph, spl, squ, tch} |
| 7 | Consonant di-graphs = {sh, ch, th, wh, ph, tw, gh, ck, wr, ng}. |
| 8 | Short vowel sound → If number of vowels in word=1. |
| 9 | Long vowel sound → if number of vowels in word > 1. |
| 10 | Double final consonant → if the word has same double final consonants (--CC) |
| 11 | wordlen = number of characters in word |
| 12 | Affix = Prefix + Suffix |
| 13 | Prefix = {un-, in-, im-, dis-, a-, re-, over-, dis-, de-, out-, mis-} |
| 14 | Suffix = {-ed, -able, -less, -ive, -y, -ful, -al, -ly, -wise, -wards, -er, -ness, -or, -ion, -ation, -ment, -ist, -ery, -ity, -en, -ize, -ise, -ing} |

A pictorial representation of the Word Classification Algorithm is given in Figure 6.2.

We input the word whose class is to be determined into the algorithm. The word length of the word is found by counting the number of characters in the word. If the wordlen is less than 3, then the word is ‘simple’. If the wordlen is equal to 3, then we check if the word has ‘y’ as its last letter. If so, then we further check if the word has ‘ay’ as its last letters, in which case it is ‘simple’ else it is classified as ‘common’. On the other hand, if wordlen of the word is more than 3 characters, then the word is checked for being a contraction. To do this, the algorithm refers to the ‘compiled database’, the details of which are given in section 4.2.2.1.1 in Chapter 4.

If the word is a contraction, then it is classified as ‘Common’ as per the NAPLAN rubric. However, if the word is not a contraction, then the next step is to find the number of syllables (SC) in the word, by referring to the SyllableCount database. The details of this database are provided in section 4.2.2.1.1 in Chapter 4. Depending on the SC value, appropriate rules are applied in order to classify the word.

If SC value is 1, then the algorithm refers to the rule base for SC = 1, which is detailed in Figure 6.3. As a result of referring to the rule base, the algorithm can determine the class of the word and displays this as the output.

In the rule base for SC = 1 given in Figure 6.3, the first step is to check the ending characters of the word. NAPLAN rubric states that if the word has the particular word endings as stated, then it should be automatically classified as common. If the word endings do not match the given ones, then we check if the wordlen is less than or equal to 5. If true then Case ‘S’, detailed in Figure 6.4, is checked.

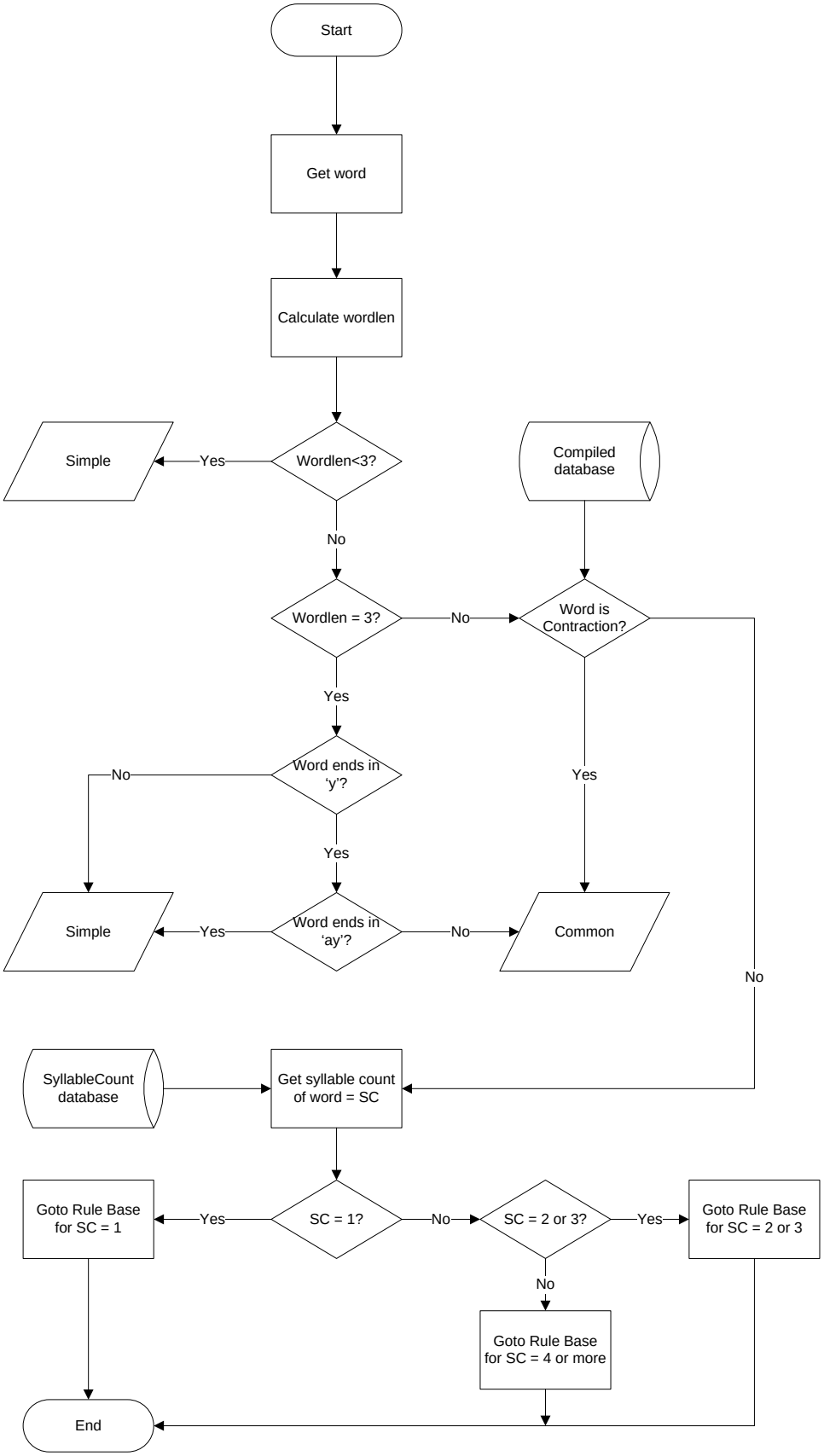



**Figure 6.2: Pictorial representation of the Word Classification Algorithm**

```
If word ends in ‘ould’ OR ’ey’ OR ’ough’ OR ‘ught’ then 'Common'. Else
        if (wordlen≤5) then go to Case ‘S’. Else
                if word=base word + affix then ‘Common’ Else
                        if (wordlen≤6 )&&(word starts with q/u/v/w/y/z) then ‘Difficult’
                        Else go to Case ‘C’.
                        If Case ‘C’ is false then ‘Difficult’
```

**Figure 6.3: Rule base for SC = 1**

If Case ‘S’ is false, then the algorithm checks if the word has an affix such that the base word is appended to the affix, in which case the word is ‘common’. Else the algorithm further checks if wordlen is less than or equal to 6 and the word starts with some specific characters. If so, then word is ‘difficult’ else if wordlen is less than or equal to 6, then the word is ‘common’. In all other cases, Case ‘C’ is referred. If case ‘C’ is not true, then the word is classified as ‘difficult’. Hence a word with one syllable is classified as either ‘simple’, ‘common’ or ‘difficult’.

```
If (8 is true)
        If Case ‘C’ is true then ‘Common’ else ‘Simple’
OR (Number of 7 in word=1)
OR (Number of 4 in word=1)
OR (Number of 5 in word = 1)
OR (10 is true)
OR (9 is true then check if word has ‘oo’/’ee’ then ‘Simple’ else ‘Common’)
then 'Simple'.
```

**Figure 6.4: Rule base for Case ‘S’**

In Case ‘S’, the algorithm checks if the word has a short vowel, that is, the number of vowels in the word is 1. In such a case, the algorithm further refers to Case ‘C’ in order to see whether the word is common. Case ‘C’ is detailed in Figure 6.5 below, where it is determined if:

- the number of consonant di-blends of type 1 and consonant digraphs in the word is 1 each OR
- the number of consonant di-blends of type 1 and consonant di-blends of type 2 in the word is 1 each OR
- the number of consonant di-blends of type 1 is 2 OR
- the number of consonant di-graphs in the word is 2 OR
- the number of consonant di-blends of type 2 and consonant di-graphs in the word is 1 each OR
- the number of consonant tri-blends in the word is at least 1

To determine the above, points 1 to 10 mentioned earlier in Table 6.1, are referred to.

```
If [(No. of 4 in word = 1) AND (No. of 7 in word = 1)]
OR [(No. of 4 in word =1) AND (No. of 5 in word = 1)]
OR (No. of 4 in word =2) OR (No. of 7 in word = 2)
OR [(No. of 7 in word = 1) AND (No. of 5 in word = 1)]
OR (Number of 6 = 1 OR 2)
then 'Common'
```

**Figure 6.5: Rule base for Case 'C'.**

If any of the conditions of case 'C' are satisfied, then the word is classified as 'common', else the algorithm control goes back to its last referral point. Currently, the last referral point is within Case 'S'. In Case 'S', the algorithm further checks if

- the number of consonant di-graphs in the word is 1 OR
- the number of consonant di-blends of type 2 in the word is 1 OR
- the number of consonant di-blends of type 1 in the word is 1 OR
- the word has same double final consonants OR

- the word has a long vowel and has the same double vowel

If any of the above conditions are satisfied, then the word is classified as ‘simple’ else if the word has a long vowel but not the same double vowel, then the word is classified as ‘common’. In all other cases, the algorithm control goes back to the last referral point in the flow.

If SC value is 2 or 3, then the algorithm refers to the rule base for SC = 2 or 3, which is mentioned in Figure 6.6. In the rule base for SC = 2 or 3, the first step is to check if the word is compound. We do this by checking the compiled database. If the word is found in the database, then we check if the wordlen is less than or equal to 10 to classify the word as ‘common’ else the word is classified as ‘difficult’. In case the word is not found, then the next step is to check if it is found in homophones. If the word is found in the compiled database, then we further check if wordlen of the word is less than 6 to classify it as ‘common’, else the word is classified as ‘difficult’. On the other hand, if the word is not a homophone, then we check if the word has the suffix ‘-ing’. Then we further check if the word has a prefix, in which case, the word is classified as ‘difficult’, else we check if wordlen of the base word is at least 8 in order to classify the word as ‘challenging’, else the word is classified as ‘common’. However, if the word does not have the suffix ‘-ing’, then we check if the word has any other affix such that the word is the base word appended with the affix. If so, then we check if Case ‘A’ is satisfied. If so, then we further check if wordlen of the base word is at least 7, in order to classify the word as ‘difficult’, else the word is classified as ‘common’. In contrast, if Case ‘A’ is not satisfied, then we check if SC = 2 and wordlen is less than or equal to 6. If these conditions are

```
If word is Compound then
        If wordlen ≤ 10, then 'Common' Else 'Difficult'.
Else if word is Homophone then
        If wordlen < 6 then ‘Common’ Else ‘Difficult’
Else if word has suffix '-ing' then
        if word has prefix then ‘Difficult’ else
                if wordlen (base word) ≥ 8 then ‘Challenging’ else ‘Common’
Else If word = base word + affix then
        if Case ‘A’ is true then
                if (wordlen (base word) ≥ 7) then ‘Difficult’ else ‘Common’
        Else if (SC = 2) && (wordlen ≤ 6) then
                If word starts with q/u/v/w/y/z then ‘Difficult’ else ‘Common’
    Else ‘Difficult’
```

**Figure 6.6: Rule base for SC = 2 or 3.**

satisfied, then we further check if the word starts with certain characters, in order to classify the word as ‘difficult’, else the word is classified as ‘common’. In all other cases, the word is classified as ‘difficult’. Hence, if the number of syllables in a word is 2 or 3, then our algorithm classifies it as either ‘common’, ‘difficult’ or ‘challenging’.

```
for every  possible affix in the word, do
        derive base word (base word = word - affix)
                check if base word + 'e' = dictWord OR base word - last consonant(base word) = dictWord OR ((base word - 'i') + 'y') = dictWord
                        if base word = dictWord then base word = True else False
        End
End
```

**Figure 6.7: Rule base for Case ‘A’.**

The algorithm refers to the affix list provided in point 12 in table 6.1 and extracts the base word of a word by stripping the affix. Hence, it is essential to check if the base word derived is actually a proper word. To do this, we use case 'A' mentioned in Figure 6.7, where we check if the derived base word is present in the SyllableCount database. Let us call the SyllableCount database word as dictWord. Then, for a given word, we first derive the base word for every possible affix that is found in the word.

According to English language conventions, before appending a suffix, some base words have to:

- drop the final 'e' (for example: have → having) OR
- add the same final consonant (for example: run → running) OR
- change the final 'y' to 'i' (for example: happy → happily)

In order to formulise this and to check if the derived base word is a proper dictWord, we check if the base word appended with 'e' is a dictWord OR base word minus the last consonant of base word is a dictWord OR base word minus 'i' then appended with 'y' is a dictWord. If either of these conditions is satisfied, then the base word is regarded as a proper word and case 'A' returns a value of true, else it returns a value of false.

In all other cases where SC value is 4 or more, the rule base for SC >= 4 or more, given in Figure 6.8 is referred to by the algorithm. In the rule base for SC of at least 4, the first step is to look up the compiled database to see if the word is compound. If so, the word is classified as 'difficult', else we check if the word has certain final characters as given in Figure 6.8. If so, then we further check if the word has a prefix, in which case, the word is classified as 'challenging', else the word is classified as 'difficult'. In case the word does not contain the final characters as required, then we check if the word has an affix such that the word is a base word appended with the affix. If so, then we check if case 'A' is satisfied. Further, we

check if the base word ends in the final characters mentioned in Figure 6.8, in which case, the word is classified as ‘challenging’, else it is classified as ‘difficult’. In all other cases, the word is classified as ‘challenging’. Hence, if a word has at least 4 syllables, then the algorithm classifies it either as ‘difficult’ or ‘challenging’.

```
If word is Compound then ‘Difficult’ else
        If word ends in tion, sion, ture, ible/able, ent/ant, ful, then
                if word has prefix then ‘Challenging’ else ‘Difficult’
        Else if word=base word + affix then go to Case ‘A’.
                if base word ends in ‘e’ OR ‘c’ OR ‘l’ then ‘Challenging’ else ‘Difficult’
            Else ‘Challenging’
```

**Figure 6.8: Rule Base for SC ≥ 4**

The word classification algorithm was coded completely in the Java language. Each rule was given a number internally and when the word class was displayed, the rules satisfied were also displayed, which gives an understanding regarding the flow of the program and how well it is able to capture the rubric. A few sample outputs are given in the next sub-section.

### 6.3.1. Sample outputs

A sample output obtained from the program is given in Figure 6.9.

```
Word = best
================================================================================
WordClassificationTest.java:52 - best is ‘Simple’
====================Rules Trace=================================================
-- 4--7--
================================================================================
```

**Figure 6.9: Word Classification Algorithm output for word ‘best’**

In the above example, the class of the word ‘best’ is to be determined. The program displays that the class is ‘simple’ which is correct according to the NAPLAN rubric. The ‘rules trace’

shows the details of rules that are satisfied while the program goes through the rules stated in the algorithm. The word ‘best’ has wordlen > 3 and is not found in contractions, hence the SC value is determined. Since SC = 1, rule base for SC = 1 is referred to (denoted as 4 in ‘rules trace’). The word does not have the word endings as stated therein, hence the next step is to check case ‘S’. In Case ‘S’, the rule ‘number of 5 in word = 1’ is true (denoted as 7 in ‘rules trace’) because ‘st’ is present in ‘best’. Hence, the result ‘simple’ is returned from Case ‘S’ and displayed by the program.

Consider the example in Figure 6.10.

```
Word = hesitation
==========================================================================
WordClassificationTest.java:94 - hesitation is ‘Difficult’
====================Rules Trace=============================================
-- 14--21--
============================================================
```

**Figure 6.10: Word Classification Algorithm output for ‘hesitation’**

In the above example, the class of the word ‘hesitation’ is to be determined. The program displays that the class is ‘difficult’, which is correct according to the NAPLAN rubric. The ‘rules trace’ shows the details of rules that are satisfied while the program goes through the rules stated in the algorithm. The word ‘hesitation’ has wordlen > 3 and is not found in contractions. Hence the SC value is determined. The SC value is 4 because there are 4 syllables in the word (he_si_ta_tion). Hence, the rule base for SC ≥ 4 is referred to (denoted as 14 in ‘rules trace’). The word is not a compound, hence the next rule – checking if the word has certain word endings (*tion, sion, ture, ible/able, ent/ant, ful*) is addressed. The rule is satisfied (denoted as 21 in ‘rules trace’) because the word ‘hesitation’ ends in ‘*tion*’. Hence, the next step is to check if the word has prefix, which is not true because the word ‘hesitation’ has the

suffix ‘-ion’ but does not have a prefix. Hence, the result ‘difficult’ is returned and displayed by the program.

Consider the example in Figure 6.11.

```
Word = baulk
================================================================================
WordClassificationTest.java:94 - baulk is ‘Common’
====================Rules Trace=================================================
--4--9--
--------------------------------------------------------------------------------
```

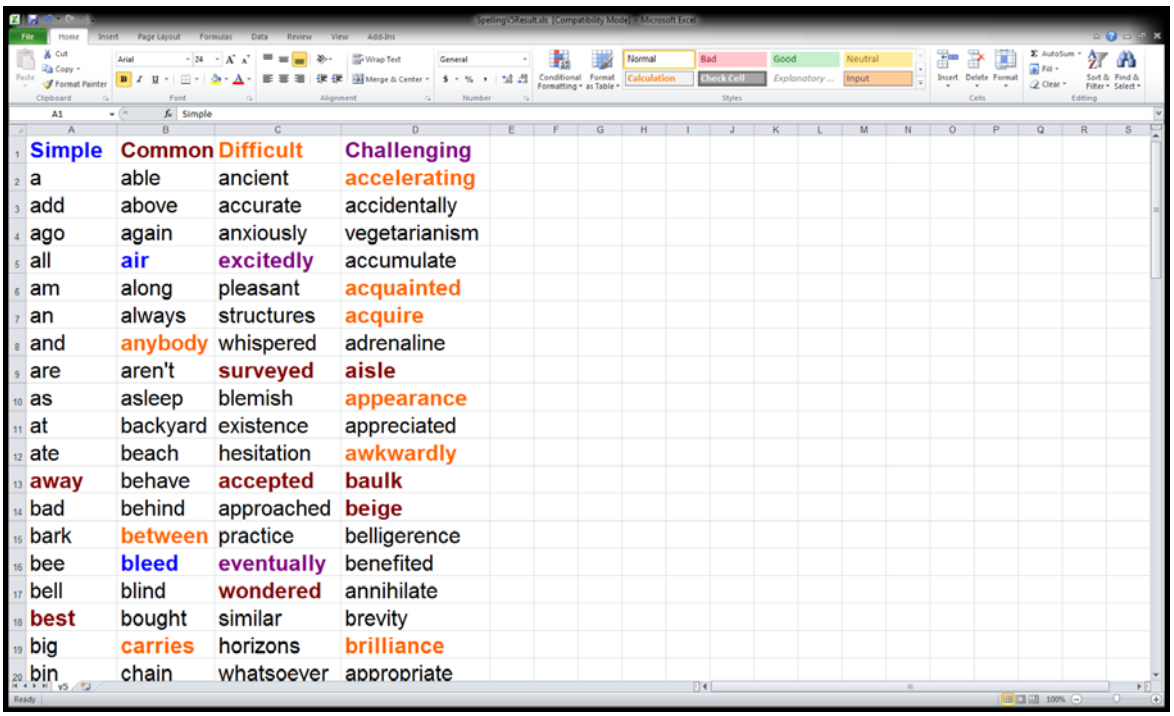
**Figure 6.11: Word Classification Algorithm output for the word ‘baulk’**

In the above example, the class of the word ‘baulk’ is to be determined. The program displays that the class is ‘common’ which is incorrect according to the NAPLAN rubric. The correct classification of this word is ‘challenging’. The ‘rules trace’ shows the details of rules that are satisfied while the program goes through the rules stated in the algorithm. The word ‘baulk’ has wordlen > 3 and is not found in contractions. Hence the SC value is determined. Since SC = 1, rule base for SC = 1 is referred to (denoted as 4 in ‘rules trace’). The word does not have the word endings as stated therein, hence the next step is to check case ‘S’. In case ‘S’, the rule ‘If 9 is true, then check if word has ‘oo’/’ee’ then ‘simple’ else ‘common’’ is true (denoted as 7 in ‘rules trace’) because the word ‘baulk’ has a long vowel but does not have the same double vowel. Hence, the rule is satisfied and the result ‘common’ is returned by Case ‘S’ which is subsequently displayed by the program. But since the classification is incorrect, we can check where the rules need tweaking and accordingly do it, to make the algorithm perform better.

In the next section, we performed testing on a comprehensive dataset of 700 randomly chosen words, in order to check the performance of the algorithm.

### 6.3.2. Testing and Results

In order to test the word classification algorithm, we compiled the test dataset as follows. We randomly selected a total of 700 words, some from the NAPLAN word lists and some from the words in the NAPLAN writing guide, exemplars section. We did this because the classes of these words are already published, so we can correctly check if the class assigned by our program is same as the one expected. The chosen 700 words comprised of 175 words each from classes simple, common, difficult and challenging. The results of the testing are provided in Figure 6.12 (please view in colour for clear interpretation).



| Simple | Common | Difficult | Challenging |
|---|---|---|---|
| a | able | ancient | accelerating |
| add | above | accurate | accidentally |
| ago | again | anxiously | vegetarianism |
| all | air | excitedly | accumulate |
| am | along | pleasant | acquainted |
| an | always | structures | acquire |
| and | anybody | whispered | adrenaline |
| are | aren't | surveyed | aisle |
| as | asleep | blemish | appearance |
| at | backyard | existence | appreciated |
| ate | beach | hesitation | awkwardly |
| away | behave | accepted | baulk |
| bad | behind | approached | beige |
| bark | between | practice | belligerence |
| bee | bleed | eventually | benefited |
| bell | blind | wondered | annihilate |
| best | bought | similar | brevity |
| big | carries | horizons | brilliance |
| bin | chain | whatsoever | appropriate |

**Figure 6.12: Results from the Word Classification Algorithm**

In Figure 6.12, the classes simple, common, difficult and challenging are colour coded in blue, maroon, orange and purple, respectively. The top row in the figure denotes the name of the class. The words under each class are positioned according to the actual NAPLAN classification of the words. Our program displays the word in black if the result obtained from the program is the same as the expected result. Otherwise, the word is displayed in the colour ac-

cording to the class obtained in the result. For example, the word ‘add’ is displayed in black because our program correctly classified it as simple. For example, the word ‘away’ is in the maroon colour in the result because our program classified it incorrectly as a common word. Similarly, the word ‘eventually’ is displayed in purple because although it should be classified as difficult, our program classified it as challenging. The complete results obtained for the 700 words can be found in the Appendix B.

### 6.3.3. Performance Evaluation

In order to analyse the performance of the system and because our algorithm is fundamentally a classification model, we use the performance metrics of precision, recall and F-measure, as mentioned previously in chapter 5. To calculate the precision, recall and F-measure, the first step is to determine the true positives (TP), false positives (FP) and false negatives (FN) for each class. Then, using the formulae mentioned earlier in Chapter 5, section 5.3.4, we calculate the metrics specific to each class. Then we determine the average precision, average recall and average F-measure. All these values are given in Table 6.2 below.

**Table 6.2: Results of performance evaluation of the Word Classification Algorithm**

| Word class | TP | FP | FN | Precision | Recall | F-measure |
|---|---|---|---|---|---|---|
| Simple | 158 | 18 | 17 | 0.90 | 0.90 | 0.90 |
| Common | 143 | 56 | 32 | 0.72 | 0.82 | 0.76 |
| Difficult | 127 | 116 | 48 | 0.52 | 0.73 | 0.61 |
| Challenging | 64 | 17 | 111 | 0.79 | 0.37 | 0.50 |
| Average value | | | | 0.73 | 0.71 | 0.69 |

Overall, the performance varies somewhat between each category. The average values for precision, recall and F-measure indicate the performance of the Word Classification Algorithm. Hence, we can conclude that the Word Classification Algorithm performs with a precision, recall and F-measure of 0.73, 0.71 and 0.69, respectively, which is quite promising. That said, on closer inspection, we see that the precision of the 'difficult' class is quite low when compared to that of other classes and the recall of the 'challenging' class is very low when compared to that of other classes. Hence, the next step is to delve deeper into the errors and investigate the reasons for this, which is discussed in the next section.

### 6.3.4. Discussion of results

The precision of the 'difficult' class is 0.52 which means that out of two words that are detected as 'difficult', roughly only one is actually 'difficult'. On investigating the results obtained for the dataset, we realise that many 'challenging' words are incorrectly classified as 'difficult' by our program. This is also indicated by the low recall of only 0.37 obtained for the 'challenging' class. Furthermore, on investigating errors in classes 'simple' and 'common', we state the following reasons.

1. Word classification according to NAPLAN rubric is based on:
    a. certain language-based heuristics and rules
    b. the frequency of word in everyday usage (according to common sense)
    c. the correlation between the spelling of the word and its pronunciation key
    d. the relationship between the usage frequency of the word and the correlation
2. **No usage frequency data**. The major limiting factor for us is not having access to the usage frequency of the word. In fact, if we can make use of such a database, then the accuracy of our algorithm would be much greater than 80%. The incorrect classifica-

tions are in the adjacent classes most often. This is because the higher the usage frequency of a word, the lower its class. Hence, most frequently used single syllabic words would be either ‘simple’ or ‘common’. On the other hand, words which have the lowest usage frequency are classified into the highest class, that is, ‘challenging’. Hence, words such as ‘aisle’, ‘beige’, ‘brusque’, ‘camouflage’ and ‘thermonuclear’ are ‘challenging’, but since we don’t have the usage frequency data, our algorithm classifies these words as either ‘common’ or ‘difficult’.

3. **Minor inconsistencies** in word classification rules, laid down by the NAPLAN rubric. Consider the following examples.

    a. The rubric states that*: Words ending in final double consonants are to be classified as ‘simple’ (For example, ‘will’, ‘less’).* But ‘wall’, ’tall’, ‘small’ and ‘class’ are classified as ‘common’ and ‘guess’ is classified as ‘difficult’ by the same rubric.

    b. The rubric states that*: High frequency, long vowel, single syllabic words are to be classified as ‘simple’ (For example, ‘food’, ‘feet’). B*ut then, ‘door’ and ‘green’ are classified as ‘common’ by the same rubric.

    c. The rubric states that*: Words in which the final consonant is to be doubled before adding the suffix, are to be classified as ‘common’ (For example, ‘webbed’, ‘zapped’).* But then, ‘matted’ is classified as ‘difficult’ by the same rubric.

    d. The rubric states that*: Words with suffixes where the base word is ending in ‘e’ are to be classified as ‘challenging’ (For example, ‘changeable’, ‘imagina-*

*ble')*. However, 'agreeable', 'unbelievable' and 'valuable' are classified as 'difficult' by the same rubric (maybe because of higher usage frequency).

e. The rubric states that*: Foreign words are to be classified as 'challenging'*. However, 'origami' is classified as 'difficult' by the same rubric.

f. The rubric states that*: Single syllabic words ending in 'ough' are to be classified as 'common' (For example, 'cough').* But 'bough' is classified as 'difficult' by the same rubric (maybe because of lower usage frequency).

g. The rubric states that*: Words for which adding a suffix does not change the base word are to be classified as 'common' (For example, 'adults',' sadly').* However, 'gnawed' is classified as 'difficult' by the same rubric.

4. **Restrictive lists**. For the purpose of proof of concept, we make use of a **subset** of affixes, consonant blends, consonant digraphs and consonant trigraphs. This has hindered us from detecting words which have affixes, etc. but which were not in our lists. For example, 'virologist' and 'absenteeism' are not detected as 'difficult' because we do not have '–ist' and '–ism' in our affix list. Similarly, 'crowd' is detected as 'simple' instead of 'common' because we do not have 'cr' and 'wd' in our consonant blends list. Hence, if we could use comprehensive lists, then our algorithm could classify these words successfully.

5. **Weighted entries in lists**. If the lists have weights assigned to the entries, then the performance of the algorithm can be improved further. For example, the suffix '-ly' can be assigned a higher weight than the suffix '-y' so that words such as 'particularly' and 'excitedly' can be classified correctly as 'difficult' instead of 'challenging'.

6. **Correlation not captured**. Currently, our algorithm does not capture the correlation between the spelling of the word and its pronunciation key. Hence, we have some incorrect classification results, such as

    - ‘shoulder’, ‘mucous’, ‘gnawed’, ‘quay’, ‘league’ and ‘yacht’ are classified as ‘common’ instead of ‘difficult’

7. **Relationship between correlation and usage frequency not captured**. Currently, our algorithm does not capture the relationship between the correlation (as mentioned in point 6 above) and usage frequency of the word. The relationship between them can be either direct or indirect, as follows.

    a. **Direct relationship.** When both the correlation (between spelling and pronunciation key of a word) and the usage frequency of a word are high, then the word is placed in a lower class. For example, ‘test’ is ‘simple’. Similarly, ‘imaginative’ and ‘personalities’ are ‘difficult’.

    b. **Inverse relationship.** When the correlation (between spelling and pronunciation key of a word) is low but the usage frequency of a word is high, then the word is classified in a lower class. For example, ‘islands’, ‘trouble’ and ‘although’ are ‘common’ and ‘spectacular’ is ‘difficult’.

If we assign the values high, medium, low and lowest to the correlation and similarly to the usage frequency, then word classification can be captured according to the following table.

**Table 6.3: Word Class and its respective relationship between correlation value and usage frequency**

| Class | Relationship between correlation value and usage frequency value |
|---|---|

| | | |
|---|---|---|
| Simple | High-high | Low-high |
| Common | Medium-high | High-Medium |
| Difficult | Low-high | High-Low |
| Challenging | Low-Lowest | Lowest-lowest |

8. We acknowledge that our algorithm can detect only mono-syllabic ‘simple’ words at the moment. The bi-syllabic words which are ‘simple’ words can be easily captured by including a sub-rule ‘When compound word length<=6 then ‘simple’’ within the rule of checking if word is compound, in the ‘Rule Base for SC = 2 or 3’.

9. Words which are ‘difficult’ and ‘challenging’ are most difficult to formulise and capture because these two classes consist of a myriad type of words ranging from mono-syllabic, bi-, tri- and multisyllabic. For example, ‘pray’, ‘omit’, ‘blemish’ and ‘eventually’ are all ‘difficult’ words.

10. The rules for the attachment of suffixes, as pointed out in [2], can be applied to improve the accuracy of the word classification program. Some of the rules are *replace lexeme ending ‘f’(except the case of ‘ff’) by ‘v’ if suffix starts with a vowel or ‘y’* and *remove lexeme ending ‘y’ if suffix starts with ‘i’*.

11. In the four classes of words, it is obvious that simple words are mainly most frequently used words whereas challenging words are mainly less frequently used. In other words, the more a word is used, the easier it is considered to be. Hence, the work in [3, 4] can be used to improve the classification of words by considering word frequency and word usage factors.

In the next section, spelling mark algorithm is explained in detail. It assigns a spelling score for the essay.

## 6.4. Spelling Mark Algorithm

The NAPLAN rubric for assessing spelling is coded using mathematical formulation in order to develop the spelling mark algorithm. As shown in Figure 6.13, to process an essay and assign it a mark for spelling, the first step is to determine the total number of words in the essay (W) and the number of spelling errors in the essay (S). The values of W and S are extracted from the XML file of the essay, obtained from the "SpellChecker" program, as explained in Chapter 5. If all the words in the essay are incorrect then spelling mark is 0. Therefore, if W = S then SpelMark = 0. If W is not equal to S, then the suggestion word for every spelling error from the XML file of the essay is obtained. Further, the correct words from the essay are obtained.

Next, make a list of the correct words in the essay such that each word is mentioned only once and for every word in the list, find its class. Then find the total number of 'simple' words (a), the total number of 'common' words = b, the total number of 'difficult' words = c, and the total number of 'challenging' words = d. Make a list of suggestion words such that each word is mentioned only once. For every word in this list, find its class. Then find the total number of spelling errors in the 'simple' words (ae), the total number of spelling errors in the 'common' words (be), the total number of spelling errors in the 'difficult' words (ce), and the total number of spelling errors in the 'challenging' words (de).

Then calculate the percentage of correct simple words = a% = $\left(\frac{a}{a+ae}\right) * 100$ and the percentage of correct common words = b% = $\left(\frac{b}{b+be}\right) * 100$.

*If ((a > 0) but (b = c = d = 0)) then SpelMark = 1 else*

if (a% = 100) && (b% = 100) && (*10 ≤ c < 15) && (c > ce) then SpelMark = 6 else*

*if* (a% = 100) && (b% = 100) && (*c ≥ 10) && (c > ce) && (d > 0) then SpelMark = 6 else*

*if* (a% = 100) && (b% ≥ 80) && *(c ≥ 10) && (c > ce) then SpelMark = 5 else*

*if* (a% = 100) && (b% ≥ 80) && *((c ≥ 2) OR (d ≥ 2)) && ((c > ce) OR (d > de)) then SpelMark = 4 else*

*if ((a% ≥ 80) && (b ≥ 20) && (b% ≥ 80) then SpelMark = 3 else*

*if (a% ≥ 80) && (b ≥ 2) then SpelMark = 2 else SpelMark = 1*

*End*

**Figure 6.13: Spelling Mark algorithm**

If only simple words are present in the essay, then the spelling score is 1. Then we use a top-down approach to develop the rules. To assign a score of 6, we check if the percentage of simple and common words is equal to 100 AND the number of difficult words is between 10 and 15 AND if the number of difficult words is greater than the number of errors in difficult words. On the other hand, if the number of difficult words is less than 15 but more than 10, then we check if the percentage of simple and common words is equal to 100 AND the number of difficult words is greater than the number of errors in the difficult words AND the number of challenging words is more than 0. We assign a score 6 if this rule is satisfied.

If there are no challenging words in the essay, then to assign a score 5, we check if the percentage of simple and common words is equal to 100 AND the number of difficult words is greater than or equal to 10 AND the number of difficult words is greater than the number of errors in difficult words. If this rule is not satisfied, then to assign a score of 4, we check if the percentage of simple and common words is 100 and 80, respectively AND the number of common or difficult words is at least 2 AND the number of common or difficult words should be greater than the errors in that class. On the other hand, when the percentage of both simple

words and common words is less than 100 but at least 80, then to assign a score 3, we check if the number of common words is at least 20. If not, then to assign a score 2, the number of simple words should be at least 80 and the number of common words should be at least 2. Else, the essay will be assigned a score of 1.

Coupled with the word classification program, the Spelling mark program can assign a suitable spelling grade to the essays.

### 6.4.1. Testing and Results

To determine the class of a word, the Word Classification Algorithm is used. However, to check the actual performance of the Spelling Mark algorithm, it is imperative to:

- minimize and effectively avoid the error that might be induced from the word classification algorithm. Hence, the approach we adopt is to determine the class of a word by first checking in the NAPLAN database. If the word is not found in the database, then the word classification algorithm is used for the purpose of assigning the class to the word.
- choose essays which contain words such that most of the words can be found either in the NAPLAN database or in the SyllableCount database that is used by the Word Classification Algorithm. Hence, we randomly chose about 5 essays in each score category from 0-6 and then selected those essays which satisfied the present condition. Our final test dataset consisted of 14 essays.

The complete Spelling Mark algorithm is coded in the Java language. A sample output is shown in Figure 6.14. The output shown in Figure 6.14 is for an essay from score category 2, as mentioned in the first line. Firstly, the correct words and their classes are given. The words which are found in the NAPLAN database are listed with only the class name at the head of the sentence. However, the words which are not found in the NAPLAN database are marked

'NotFound' at the head of the sentence and their classes are determined by the Word Classification program.

D:\essay grading\SpelMarkDataset\score2\ Hansen_11.doc

==========

Correct Words:

==========

Simple - one, day, i, a, dog, in, the, and, him, with, me, he, was, my, pet, name,

Common - found,

NotFound - lost:'Common', street:'Common', took:'Simple', he's:'Common', jimmy:'Common',

==========

Suggestions for Spelling errors:

==========

Common – home,

SpellingMarkTest.java: **SpelMark = 2**

**Figure 6.14: Output from Spelling mark algorithm**

Secondly, the suggestion-words for spelling errors and their classes are given. Again, the words which are found in the NAPLAN database are listed with only the class name at the head of the sentence (for example, Common –home). Finally, the program displays the Spelling mark result as '2' which is correct.

We performed such testing for the complete dataset of 14 essays and report the results.

### 6.4.2. Performance Evaluation

Since the spelling mark algorithm scores and rates an essay into one of the score categories from 0 to 6, we use the widely recognised rater agreement metrics as performance metrics. The various rater agreement metrics are defined below.

1. Perfect Agreement: When the rater achieves the exact result as the human marker, it is called perfect agreement. Ideally, the perfect agreement should be 1 for a rater.
2. Adjacent Agreement: When the rater achieves a result which is not exactly the same as the human marker but is adjacent to the mark assigned by human marker, it is called adjacent agreement and can be of two types:
    a. One-point adjacent agreement : The result given by the rater is one point less or more than the result given by the human marker
    b. Two-point adjacent agreement : The result given by the rater is two points less or more than the result given by the human marker
3. Non-adjacent agreement: When the result given by the rater is more than 2 points more or less than the result given by the human marker, it is called non-adjacent agreement.
4. Perfect+adjacent agreement: This metric is used to indicate the overall accuracy of the AEG system in terms of accordance with human markers [6]. As the name suggests, it is obtained by adding the perfect agreement value with the adjacent agreement value.

We use the performance metrics of perfect agreement and adjacent agreement, which are used extensively while reporting on the performance of an AEG system, as discussed in Chapter 2.

After a thorough manual selection process, a total of 14 essays were selected for the testing. Figure 6.15 shows that two essays were selected in each score category from 0 to 6. The results obtained for each score category are denoted in terms of perfect agreement, one-point adjacent agreement and two-point adjacent agreement. Surprisingly, and much to our relief, there were no essays which resulted in a non-adjacent agreement. In the score category '0', one essay was given the exact score of '0' by our algorithm and another was given a score of 1. In the score categories '1' and '2', all four essays were given the exact score by our algorithm. In the score category '3', one essay was given an exact score while the other essay re-

ceived a one-point adjacent score. In the score category '4', one essay was given an exact score while another essay was given a score with one-point difference.

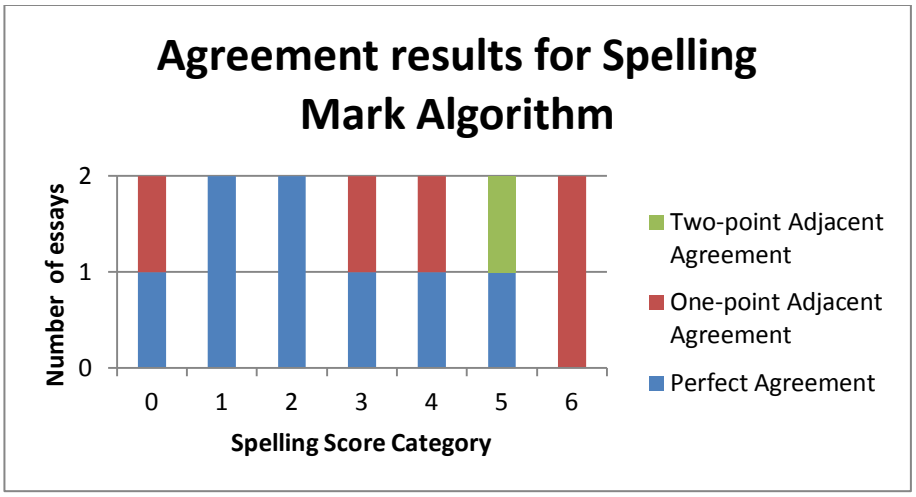



**Figure 6.15: Agreement results from Spelling Mark algorithm**

In the score category 5, one essay was given an exact score while another essay was given a score with a two-point difference. In the score category '6', both the essays were given a score with a one-point discrepancy. In all, a total of eight essays received a perfect agreement score, a total of five essays received a one-point adjacent agreement score and only one essay received a two-point adjacent agreement score, as shown in Figure 6.15 (please view in colour for clear interpretation).

Table 6.4 indicates the number of essays in each score category, the number of essays for which exact agreement was obtained and the number of essays for which adjacent agreement was obtained.

**Table 6.4: Performance evaluation of the Spelling Mark algorithm**

| | Number of essays (of n = 14) | Percentage |
|---|---|---|
| Perfect Agreement | 8 | 57.1% |
| One-point Adjacent Agree- | 5 | 35.7% |

| | | |
|---|---|---|
| ment | | |
| Perfect + Adjacent agreement | | 92.9% |
| Two-point Adjacent Agreement | 1 | 7.1% |
| Total | | 100% |

Hence, the Spelling Mark algorithm performs with a perfect agreement rate of about 57%, a one-point adjacent agreement rate of 36% and a two-point adjacent agreement rate of 7%. Although it might seem that the results are not very good, the metric of perfect + adjacent is 92.9% which is exceptionally good for the first attempt at scoring spelling according to NAPLAN rubric. In the next section, we discuss the results in detail.

### 6.4.3. Discussion of results

Of the 14 essays, the essays which were given an adjacent agreement score were investigated and the following reasons were found.

1. The Word Classification program was used to classify words which were not found in the NAPLAN database. The program detected some 'common' words as 'difficult' and some 'difficult' words as 'challenging'. Hence, the number of words in the 'difficult' class was found to be more than they actually were. Accordingly, the Spelling Mark algorithm assigned a higher score to the essay than it deserved. This was the main reason that some essays were not assigned an exact score. In contrast, essays from which almost all words were found in the NAPLAN database achieved an exact agreement score.

   In order to overcome this issue, for the future it is recommended that the classes assigned by the Word Classification algorithm be first verified by expert markers at WADET and subsequently, these words can be added to the NAPLAN database.
2. Many proper nouns like names of persons (for example, Aaron, Mathew and Ryan) cannot be found in the NAPLAN database. So, the alternative was to classify them by

the Word Classification algorithm. However, they could not be classified even by the Word Classification program because the SyllableCount database does not include such proper nouns. Ultimately, these words could not be included in the data required by the Spelling Mark algorithm. Hence, the algorithm assigned such essays a lower score than deserved.

3. In the spelling errors highlighted by Word, we noticed that proper nouns were highlighted and alternate spellings were suggested for them, for example, if the student used "Micheal" in the essay, all instances of "Micheal" were detected as spelling errors and "Michael" was suggested. Since it is entirely the student's call to use a name spelled according to his liking and since it would be unfair to grade fictitious names for 'spelling', we omitted such spelling errors. It is to be noted that spelling errors detected in all other proper nouns, such as names of places in the world and the names of recognised, famous people, were counted as errors. This can also be done by using a Name Entity Recognition code, a natural language processing module that is able to detect all proper nouns that represent names and entities. For more information on the use of NER, readers are directed to [5].

In the next section, a summary of the main points discussed in this chapter are recapped and the chapter is concluded.

## 6.5. Conclusion

In this chapter, the complete methodology of the automated marking of spelling has been explained. The working of the two algorithms, the Word Classification algorithm and the Spelling Mark algorithm that form the backbone of this module, is detailed. The Word Classification Algorithm is responsible for classifying a word into one of four classes: simple, common, difficult and challenging, according to the NAPLAN rubric. The Spelling Mark algorithm is responsible for assigning a mark automatically, according to the NAPLAN rubric. Both the algorithms are tested rigorously using datasets. The results obtained during the testing are presented and the performance of the algorithms is evaluated using widely used performance metrics. The average values of precision, recall and f-measure for the word classification algorithm are 0.73, 0.71 and 0.69. The perfect+adjacent agreement value for spelling mark algorithm is 92.9%. The performance of both the algorithms is quite good considering the fact that this is the first attempt in automating the spelling mark process for the NAPLAN rubric. In the areas where the performance was not up to the mark, the reasons for the same are discussed.

In the next chapter, the complete methodology of the automated marking of the vocabulary module is explained.

## 6.6. References


[1] (2011, 14 October). *Vowel Length*. Available: http://en.wikipedia.org/wiki/Vowel_length#Traditional_long_and_short_vowels_in_English_orthography

[2] A. Neviarouskaya, *et al.*, "SentiFul: A Lexicon for Sentiment Analysis," *Affective Computing, IEEE Transactions on*, vol. 2, pp. 22-36, 2011.

[3] P. Finn, "Computer-aided description of mature word choices in writing," in *Evaluating writing: Describing, measuring, judging*, Buffalo, NY, 1977, pp. 69-90.

[4] H. M. Breland, "Word Frequency and Word Difficulty: A Comparison of Counts in Four Corpora," *Psychological Science*, vol. 7, pp. 96-99, 1996.

[5] L. Hon Wai, *et al.*, "Towards the use of semi-structured annotators for Automated Essay Grading," in *4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, Dubai, 2010, pp. 228-233.

[6] L. S. Larkey, "Automatic essay grading using text categorization techniques," in *Proceedings of the 21st ACM/SIGIR (SIGIR-98)*, Melbourne, Australia, 1998, pp. 90-96.

# Chapter 7: Automated Scoring of Vocabulary

## 7.1. Introduction

In this chapter, the methodology and design of the vocabulary module is explained. The objective of this module is to perform automated scoring of the vocabulary criterion, according to the NAPLAN rubric. To grade vocabulary in free-text responses such as essays, it is important to develop two different approaches separately: one for grading poor essays and another for grading good essays. This is to ensure that we use and employ appropriate techniques to grade essays, depending on the level of proficiency demonstrated in the language therein.

Accordingly, in this chapter, a heuristics and rule-based algorithm for grading vocabulary in poor essays is explained after which an artificial neural network-based intelligent algorithm for grading vocabulary in good essays is outlined. The working of the algorithm for poor essays with the help of sample essays is detailed. The dataset used for the algorithm for good essays is explained as is the simulation process. Additionally, the results for both algorithms are provided in detail. Finally, the performance of each methodology is highlighted by applying a number of performance metrics widely used in the area of data mining and automated

essay grading. This is followed by a discussion of the results and a conclusion at the end of the chapter.

In the next section, the methodology of the vocabulary module is explained in detail.

## 7.2. Automated Scoring of Vocabulary

The vocabulary module constitutes two main algorithms. In this section, those two novel algorithms for grading vocabulary are presented. A heuristics and rule-based algorithm for grading vocabulary in poor essays is explained in section 7.3. Each word in the essay is considered only once and is checked to determine if it can be found in the basic vocabulary database. These words are called ‘content words’. Then, the percentage of content words is calculated and rule base is used to assign a vocabulary score to the essay.

In the section 7.4, an intelligent algorithm based on artificial neural networks to grade vocabulary in good essays is presented. A POS tagger is used to identify the part of speech of each word in the essay. Then, the number of words in different parts of speech is calculated. Other surface features of the essay are measured and finally, these are fed into the neural network model as inputs. The network is simulated to obtain the vocabulary score as the output.

In the next section, the methodology of algorithm for poor essays is elucidated with the help of an illustration.

## 7.3. Algorithm for Poor Essays

As previously mentioned, poor essays contain mainly random typing, are extremely poor in spelling and punctuation, or have words less than 80. Hence, scoring them automatically for vocabulary is a challenging task. Moreover, since there are no proper sentences in poor essays, we cannot use a part of speech tagger to identify the parts of speech of words used in the essay. Hence, the work-around which was developed is to use a database of basic vocabulary words which are nouns, verbs, adverbs, adjectives, etc.

A thorough observation of the dataset allows us to conclude that poor essays have a potential vocabulary score of 0, 1 or 2. In some extremely rare cases, there might be a score 3, but we do not take this into consideration for the same reason. This is reasonable and acceptable because given the strict NAPLAN rubric, it is nearly impossible for a poor essay to gain a vocabulary score of more than 3.

For the purpose of scoring the vocabulary in an essay, it is logical and necessary to consider each word only once, irrespective of the number of times it is repeated. We manually compiled the “Basic Vocabulary” database to be used by the algorithm, by using two word lists that are widely recognised and have been previously used in text analysis [1]. The two word lists are: Ogden’s Wordlist [2] and the ‘Voice of America’ list [3]. The Ogden wordlist was compiled by Ogden and consists of 850 of the most basic words from the English language. The Voice of America list was published in 2009 and is a more comprehensive list that contains most of the common words in the English language from A to Z. The content of these lists were described in great detail in chapter 4.

In order to grade poor essays for vocabulary, we developed the algorithm shown in Figure 7.1. It involves the following process:

1. Firstly, we find the total words in the essay (EW).
2. Then from the essay, we create a list of all unique words. This list is called *Unique*. It is important to note that *Unique* contains every word from the essay only once, irrespective of how many times it is repeated in the essay, hence the name. Then, we find the total number of words in *Unique* (denoted by 'W').
3. For every word in *Unique*, we need to check if it is present in the database.

   All the words in the essay that can be found in the database are called 'content

```
For an essay,
Get EW = Total number of words in essay.
Create a list of all words in essay such that each word is included only once. Call this list unique.
Get W = Total words in unique
Set CW = 0
For every word in unique, do
        If word is present in database
                Increment CW by 1.
Return CW
End
Calculate CW% = (CW/W)*100
        If (CW% ≥ 40) then
                If (EW ≥ 30) then Vocab Score = 2 else Vocab score = 1
        Else
                If (0 < CW% < 40) then Vocab Score = 1 Else Vocab Score = 0
        Return Vocab Score
End
```

**Figure 7.1: Algorithm for scoring vocabulary in poor essays**

The total number of content words is stored in CW. Then, the percentage of CW is calculated using the formula,

CW% = $(CW/W) * 100$

4. Finally, heuristic rules are developed and employed to assign the vocabulary score.

   According to the rubric, at least some content words are to be present in the essay for it to be assigned a score of '1'. Hence, if no content words are found, then the vocabu-

lary score will be 0. From the NAPLAN marking guide, a close analysis of the exemplars for score categories 1 and 2 lead us to conclude the following points:

- For a score 1, roughly half of the words in the essay should be content words.
- For a score 2, more than half of the words in the essay should be content words and the essay length should be more than 30 words.

We empirically derived the values of 'CW' and 'EW to be 40 and 30, respectively.

Hence, using the rules stated in Figure 7.1, we first check if CW% is at least 40. If so, then we further check if EW is at least 30, in which case the vocab score of '2' is assigned, else, a score of '1' is assigned. However, if CW% is between 0 and 40, then a vocab score of '1' is assigned. Otherwise, a vocab score of '0' is assigned because there are no content words in the essay when CW% = 0. The vocab score value is returned by the algorithm and displayed by the program.

The algorithm was implemented completely in Java language. The sample outputs obtained from the program and the results are detailed in the next section. After this, the poor essays dataset was obtained from the filter process and testing was undertaken.

### 7.3.1. Testing and Sample Results

In this section, the detailed working of the above algorithm is discussed with the help of a sample essay. A sample output obtained for a poor essay is shown in Figure 7.2. The essay name is 'Green_5', as mentioned in the first line in the output. Then, the list '*Unique*' is displayed for the essay.

In this particular essay, the number of words is the same as the number of words in *Unique*, which means that no word is repeated more than once. Then, the database matching result is displayed in *match* along with the list of words that were found in the *Basic Vocabulary* database. Since five words were matched, CW = 5. CW% is calculated and obtained as being

around 83. The *Rules Trace* in the output gives details of the rule that is satisfied by the program while processing the result. Since the rule 'CW% is ≥ 40' is satisfied, the next step is to check the EW value. As EW is less than 30, hence the rule 'If (EW ≥ 30) then Vocab Score = 2 else Vocab score = 1' is satisfied (denoted by 2 in the *Rules Trace*). Accordingly, the vocab score is assigned as '1' which is the same as assigned by human expert markers.

```
D:\Essay Grading\Poor Essays\Green_5.doc
================================================================================
Unique
================================================================================
On, the, last, day, of, scl
EW = W = 6
Match: day, last, of, on, the
CW = 5
(5 / 6)*100 = 83.33333333333334
================================================================================
Rules Trace
================================================================================
---2---
Vocab score = 1
```

**Figure 7.2: Output from algorithm for grading vocabulary in poor essays**

### 7.3.2. Performance evaluation

Using the performance metrics of perfect agreement, 1-point adjacent agreement and perfect+adjacent agreement, the performance of the algorithm was evaluated. Of the 135 poor essays, perfect agreement was obtained for 88 essays as shown in table 7.1. Complete results can be found in Appendix D. One-point adjacent agreement was obtained for 47 essays which highlights that the score assigned by the algorithm was one point more or less than the actual score for the essays. Accordingly, the percentage of perfect agreement and one-point adjacent agreement values for the algorithm are 65.2% and 34.8%.

**Table 7.1: Performance evaluation of algorithm for poor essays**

| | Number of essays (of n = 135) | Percentage |
|---|---|---|

| | | |
|---|---|---|
| Perfect Agreement | 88 | 65.2% |
| One-point Adjacent Agreement | 47 | 34.8% |
| Perfect + Adjacent Agreement | | 100% |
| Total | | 100 |

In the next sub-section, a discussion of the performance of the algorithm is presented along with the reasons for the discrepancy between the actual score of the essays and the score assigned by the algorithm.

### 7.3.3. Discussion of results

After a thorough investigation to find the reasons for errors in the program results, we conclude the following.

The algorithm finds the content words in an essay, that is, the words which can be found in the *Basic Vocabulary* database. Since only correctly spelled words can be found in the database, it so happens that words that have spelling errors are omitted. However, the NAPLAN rubric considers these words as well for assigning the vocabulary score. Moreover, a human being can identify the word that the student intended to write, however, it is quite difficult for a machine to predict the same. However, we anticipate that by using the first suggestions for incorrectly spelt words, given by Microsoft Word and including them in the list *Unique*, we might improve performance even more.

Many proper nouns like names of persons (for example, Tim, Amy and Rick) cannot be found in the *Basic Vocabulary* database. So, these words could not be included in the CW% data required by the algorithm. Hence, the algorithm assigned such essays a lower score than deserved. It might appear that we could use the Noun Entity Recognition tool to detect proper

nouns and then leave them out of the algorithm. However, an NER tool would also tag a proper place such as ‘park’ as noun. Hence, skipping all nouns will not be a proper solution.

In the next section, algorithm for grading good essays is explained in detail.

## 7.4. Algorithm for Good essays

According to the NAPLAN rubric, the various parts of speech of the words in the essay are examined to grade the essay in relation to vocabulary. Additionally, good essays are those which are detected as neither anomalous nor poor. A distinctive feature of a good essay is that it has sentence formation, thus enabling us to use the POS tagger to obtain the parts of speech for each word in the essay. A detailed sample of the output obtained from the POS tagger is explained in section 4.2.2.3.1 in chapter 4. The output obtained from the POS tagger is in the form of Penn Treebank tags for each word in the essay.

We use coarse tags to collectively represent a group of related tags. For instance, the various forms of verbs such as present tense, present participle and past participle are denoted as VB, VBG and VBN, respectively, by the tagger. These tags and the tags VBP, VBZ, MD and VBD are represented using the coarse tag ‘V’ denoting ‘Verb’. By using the coarse tag ‘V’, all forms of verbs are represented together, thus making it simpler to determine the number of verbs in the essay. The complete lists of Penn Treebank tags and the coarse tags that represent them are shown in Table 7.2. We adapt this from [1] and modify it marginally to suit our requirements.

**Table 7.2: Coarse tags, their description and the Penn Treebank tags they represent**

| Coarse Tag | Description | Penn Treebank Tags |
|---|---|---|
| DET | Determiner | DT, PDT |
| ADJ | Adjective | JJ, JJR, JJS |
| N | Noun | NN, NNS, NP, NPS, PRP, FW |
| ADV | Adverb | RB, RBR, RBS |
| V | Verb | VB, VBN, VBG, VBP, VBZ, MD,**VBD** |
| WH | Words beginning with 'Wh' | WDT, WP, WP$, WRB |
| **INP** | **Interjection and Preposition** | **IN, RP** |

In Table 7.2, the tags which are in bold are introduced in this thesis, whereas all the other tags are adapted from [1]. The coarse tag 'DET' which denotes 'determiner' comprises two tags: DT (determiner) and PDT (pre-determiner). The coarse tag 'ADJ' which denotes the 'adjective' part of speech comprises three tags: JJ (adjective), JJR (comparative adjective) and JJS (superlative adjective). Similarly, coarse tag 'N' which denotes the 'noun' class part of speech comprises six tags: NN (common noun), NNS (plural noun), NP (proper noun), NPS (plural proper noun), PRP (personal pronoun) and FW (foreign word). The coarse tag 'WH' is used to denote words beginning with 'Wh' and comprises four Penn Treebank tags – WDT (Wh-determiner), WP (Wh-pronoun), WP$ (Wh-possessive pronoun) and WRB (Wh-Adverb). The coarse tag 'INP' is used to denote the 'interjection and preposition' classes of parts of speech. It comprises only two tags: IN (interjection, conjunctions and prepositions) and RP (particle).

We determine the number of unique words in each coarse tag. Unique implies that each word is counted only once, irrespective of the number of times it is used in the essay. However, if

the same word is used more than once but in a different part of speech each time, then we capture such usage as well. For example, consider the two sentences below:

1. Please ring the bell (in this context, 'ring' is a verb)
2. I bought you a ring. (in this context, 'ring' is a noun)

In the first sentence, the word 'ring' is used as a verb. However, in the second sentence, the same word is used as a 'noun'. Since we consider nouns and verbs as two separate coarse tags in the algorithm, we can capture both instances of the word. Moreover, if a person's vocabulary is good, he can use the same word in more than one way, as shown in the sentences above.

The complete algorithm for grading vocabulary in good essays is illustrated in Figure 7.3. The first step is to split the essay into individual sentences because the POS tagger takes as input one sentence at a time. For this purpose, we employ the 'Sentence Segmentation' tool available from [4]. The next step is to input each sentence to the Stanford POS tagger in order to obtain the part of speech tags for each word in the sentence. In the next step, we determine the class of each word and subsequently, the total number of unique words in each class. We use the NAPLAN database of word classification to find the class of the word and we count each occurrence of the word only once, irrespective of its frequency in the essay. From this step, the outputs obtained are:

- the total number of unique words in the class 'simple'
- the total number of unique words in the class 'common'
- the total number of unique words in the class 'difficult'
- the total number of unique words in the class 'challenging'

In the next step, we determine the type-token ratio of words in each class determined in the previous step. The type-token ratio (TTR) is a measure of diversity in vocabulary within a written text or a person's speech [5]. The formula to calculate type-token ratio is:

Type - token ratio of words in class 'simple' = $\left(\frac{a}{b}\right) * 100$

- where a = the total number of unique words in the class 'simple' and
- b = sum of (frequencies of each word in class 'simple') = $\sum_{i=0}^{n} f(wordi)$

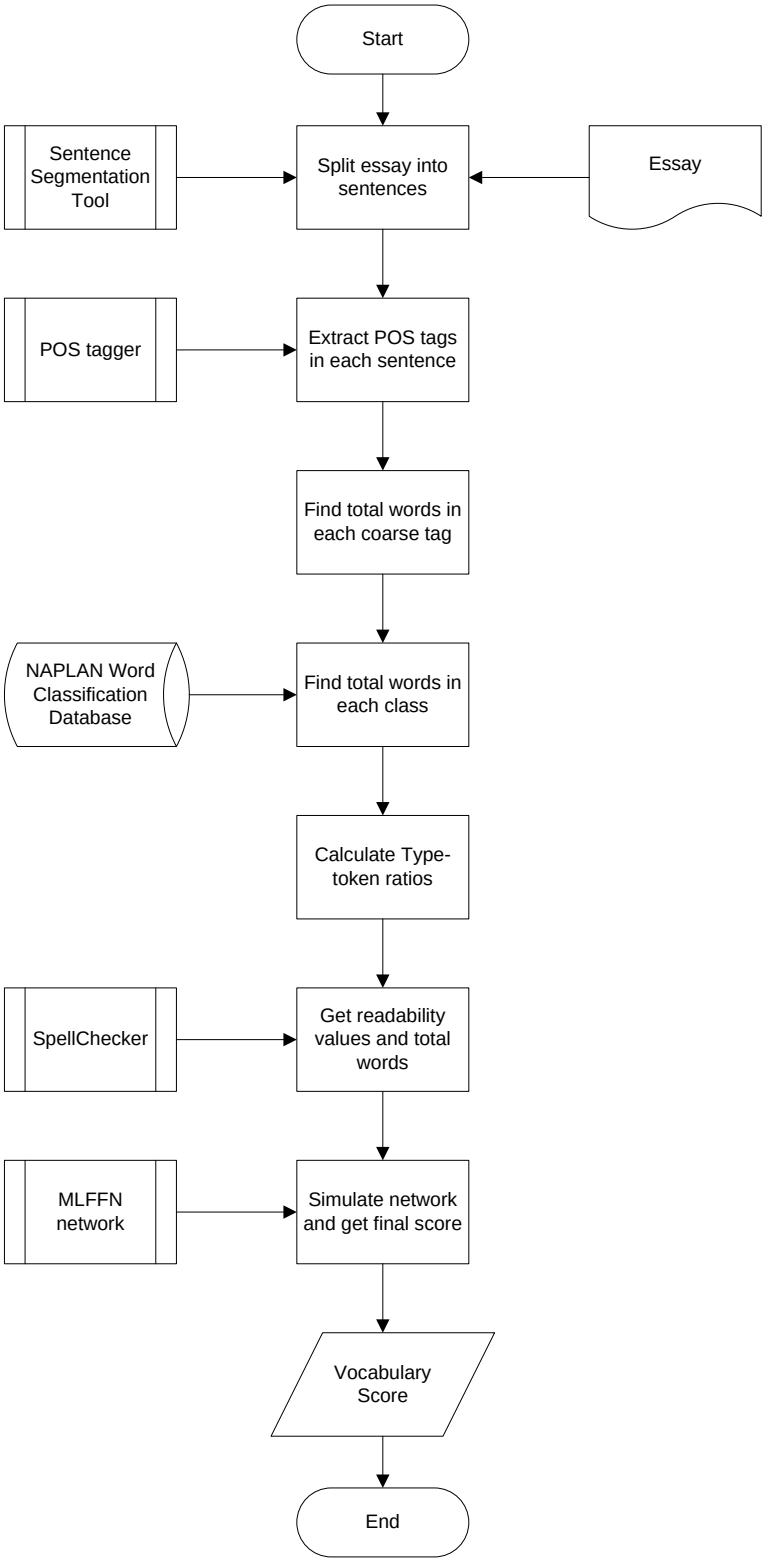



**Figure 7.3: Algorithm for grading vocabulary in good essays**

Similarly, the type – token ratio of words in the classes ‘common’, ‘difficult’ and ‘challenging’ is also calculated. Hence, there are four outputs from this step.

- Type-token ratio of words in the class ‘simple’
- Type-token ratio of words in the class ‘common’
- Type-token ratio of words in the class ‘difficult’
- Type-token ratio of words in the class ‘challenging’

In the next step, we obtain the values of the total words in the essay, the average word length, Flesch reading ease and Flesch K-grade level. The first two features have been used previously for grading essay scores in which two widely used metrics denote the readability level of a text [6]. Moreover, according to [7], the average word length is an important indicator of variety and vocabulary complexity in the essay. We obtain all these values automatically by using the ‘SpellChecker’ program, which derives the values from Microsoft Word. Hence, from this step, the outputs are:

- total words in essay
- average word length
- Flesch reading ease value for the essay
- Flesch K-grade level value for the essay

Outputs obtained from each step of the algorithm described above are used in the next step of the algorithm. In the next step, a neural network is designed and calibrated to perform automated grading of vocabulary.

### 7.4.1. Neural Network Calibration

To choose the optimal neural network, networks of various configurations are developed and simulated using the dataset. With the MATLAB GUI tool for neural networks, ‘nntool’, Multi-Layer Feed Forward Neural (MLFFN) networks of various configurations are designed.

We configured three different MLFFN networks with two layers in the hidden layer for each network. The first network had 40 neurons in the hidden layer. Then, we increased the number of neurons by 5, so the second network has 45 neurons in the hidden layer. We performed the training and calibration of this network in the same way as for the first network. Then, we increased the number of neurons in the hidden layer by 5. Hence, the third network has 50 neurons. For each network, the training algorithm is ‘trainlm’ and the learning algorithm is ‘back propagation’. The learning functions for both hidden layer 1 and hidden layer 2 is ‘tansig’ because this is a pattern classification task. Figure 7.4 shows the GUI of the MATLAB neural network training tool called ‘nntraintool’. It shows the design of the neural network consisting of the input layer which is responsible for feeding the inputs to the hidden layer. In the hidden layer, the learning function is ‘tansig’, the training algorithm is ‘trainlm’, the performance function for the network is ‘mse’ and the data division is set at random, as mentioned earlier. In the next hidden layer, the result obtained from the network is converted back to the format of the expected results by using the ‘tansig’ function as the learning function. This output is sent to the output layer as the result obtained from the neural network. For training the network, all the default settings of training parameters are used. Training stops when validation checks are performed six times.

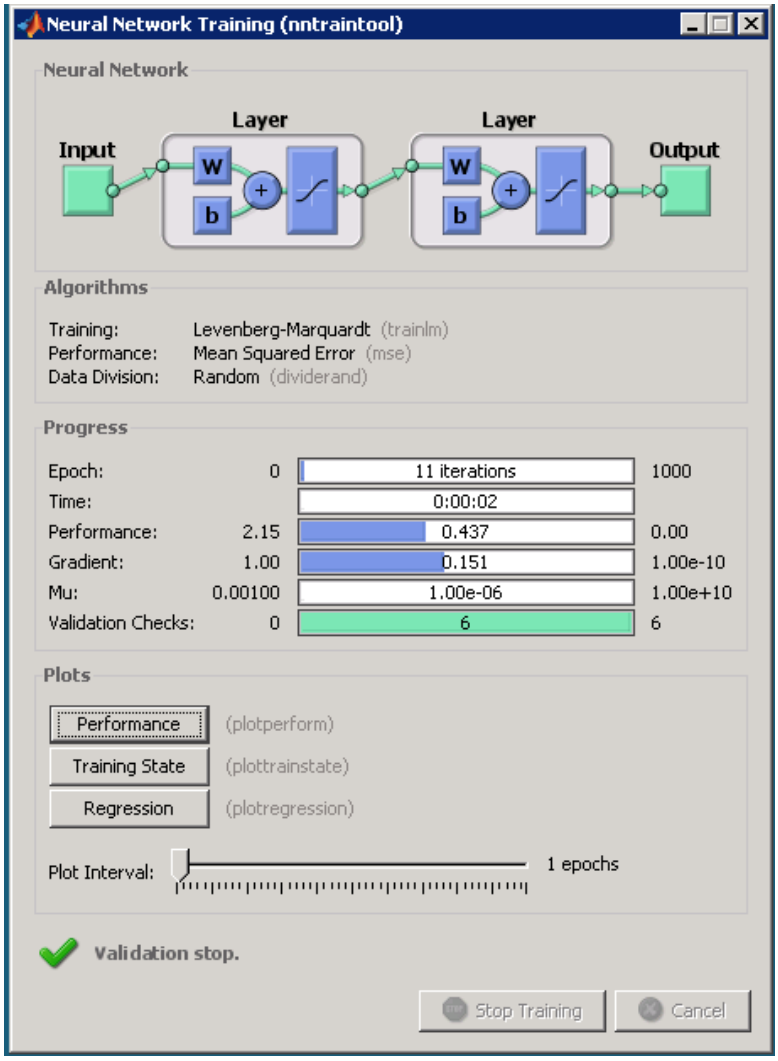



**Figure 7.4: Screenshot for network architecture and training performance.**

Using the validation technique of ‘stratification’, the essay dataset was divided manually into two sets: a training set and a testing set. We use the technique called "proportional assignment" where we choose category cut offs to put the correct number of training essays into each grade [8].

Training set comprising 80% of data is used for the training and calibration of the network. A testing set comprising 20% of data and will be used to test the performance of the network. It is important to note that the network has not seen the testing set before. Hence, it is used only after completing the training phase of the network. This is also called the ‘Hold-Out method’

in data mining. The complete dataset provided by WA-DET (Department of Education and Training, Western Australia) consists of 172 good essays. Hence, using the above technique, the training set consists of 138 essays and the testing set consists of 34 essays.

When the training set is given to the neural network, internally, the network splits it into three subsets: a training subset, a validation subset and a testing subset. This is done randomly and the data is split in the ratio of 70%, 15% and 15% to obtain the three subsets. Accordingly, the training set of 138 essays was split into a training subset consisting of 84 essays, a validation subset and a testing subset, each consisting of 27 essays.

Using the GUI shown in Figure 7.5, the data for the training phase is selected for the neural network. The inputs and targets are specified and then the network is trained. The outputs of this phase are stored as training results.

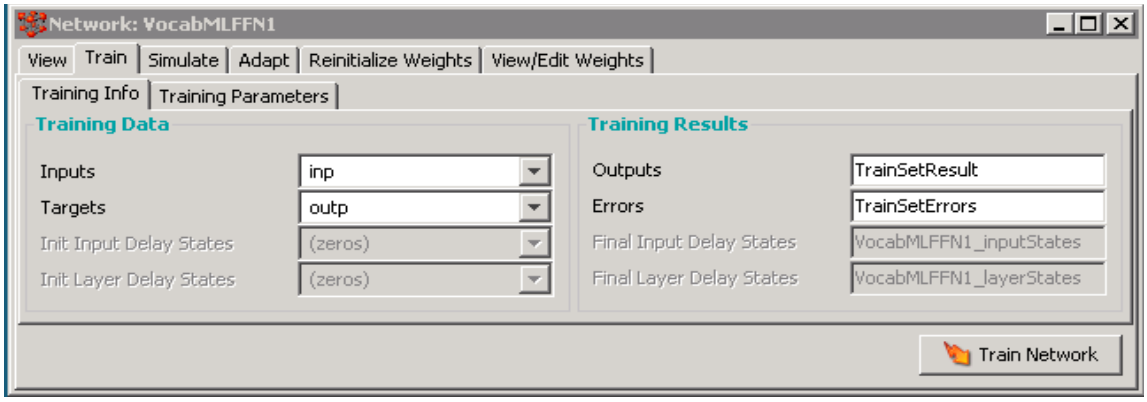



**Figure 7.5: Screenshot of the training phase of neural network**

Furthermore, the GUI in Figure 7.4 also enables us to see various plots to determine the performance of the network. The performance plot 'plotperform' shown in Figure 7.6, illustrates the best validation performance of the network. There are three curves shown in the plot. It can be seen in the plot that the MSE, during all three phases of training, validation and testing, was initially very high but reduces rapidly. The best value of MSE during training was reported as 0.72.

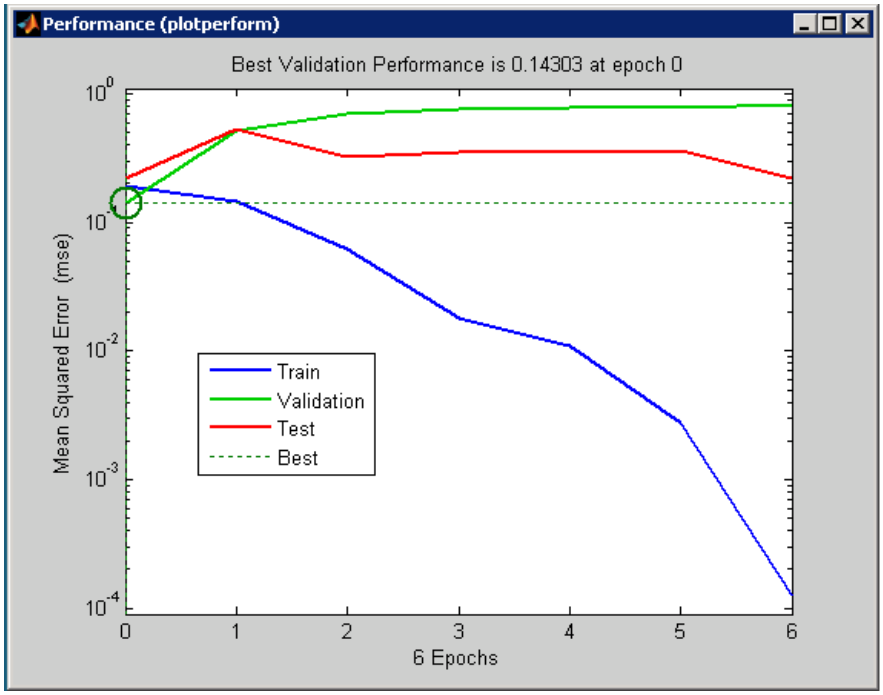



**Figure 7.6: Screenshot of lowest MSE for various phases of simulation**

For each network, training was performed using the training set. The network was calibrated by retraining over a number of iterations until it produced a low mean square error (MSE) value and until the network performed satisfactorily. Then, using the testing set, the network was simulated.

The results obtained during the training phase of each network were recorded and used to calculate the root mean square error (RMSE) was calculated using the formula as mentioned in chapter 5. The RMSE values for the training phase for different configurations of the network are 0.72 when N = 40, 0.75 when N = 45 and 0.76 when N = 50 where N = the number of neurons in the hidden layer of the neural network. The optimal performing network is the one which provided the least MSE and hence, the network with neurons = 40 is chosen as the final model.

Then this network is simulated using the testing set.

### 7.4.2. Testing, Results and Performance evaluation

Using the GUI shown in Figure 7.7, the data for the testing phase is chosen. The inputs and targets are specified for simulation and then the network is simulated. The outputs of this phase are stored as simulation results.

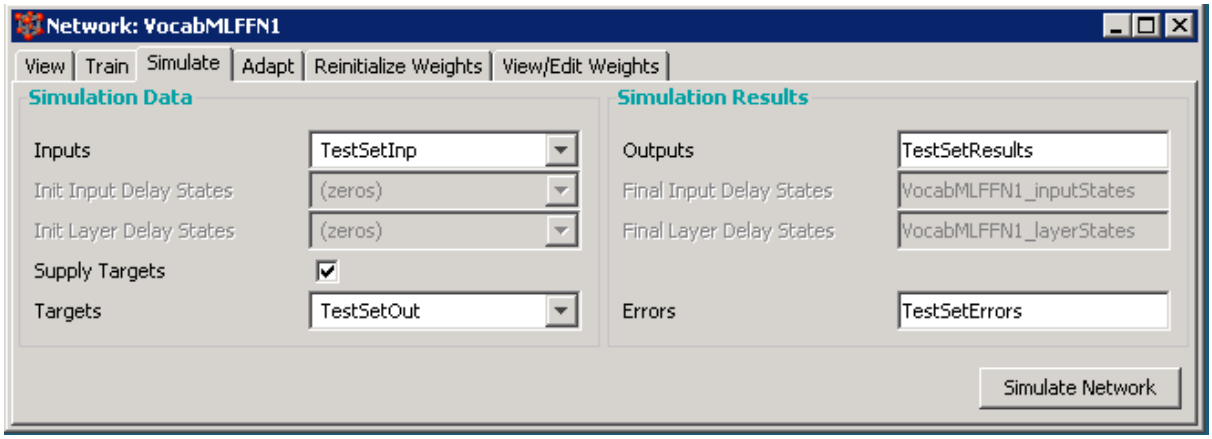



**Figure 7.7: Screenshot showing details of the testing phase**

Now, using previous performance metrics, exact agreement and adjacent agreement rate was also determined and the results are shown in Table 7.3. The vocabulary score range for good essays was from 2 to 5. The exact agreement obtained was for 19 essays, which is 55.9% and an adjacent agreement was obtained for 15 essays, which is 44.1%. Additionally, there was no two-point adjacent agreement or non-adjacent agreement for any of the essays. Further, the perfect + adjacent agreement was 100% which is very good. Hence, we can conclude that the algorithm can grade vocabulary in good essays with an overall accuracy of 100%.

**Table 7.3:Performance evaluation of the algorithm for good essays.**

| | Number of essays (of n = 34) | Percentage |
|---|---|---|
| Perfect Agreement | 19 | 55.9% |
| One-point Adjacent Agreement | 15 | 44.1% |
| Perfect + Adjacent Agreement | | 100% |

| | |
|---|---|
| Total | 100 |

The detailed results are shown in Figure 7.8. The x-axis denotes the essay number and the y-axis denotes the output score range, which is from 2 to 5 for good essays. The blue markers show the target output for the essay and the red markers show the result obtained from the algorithm. For essays where only one marker is present, this means that there is perfect agreement between targets and result. For all other essays, the discrepancy between the target and result value for the essay is shown as 1 or 2. Hence, from the figure, it is evident that there is a perfect agreement of 19 essays, and a one-point adjacent agreement for 15 essays was obtained.

In the next sub-section, a discussion is presented with the reasons for the discrepancy between the actual score of the essay and the score assigned by the algorithm.

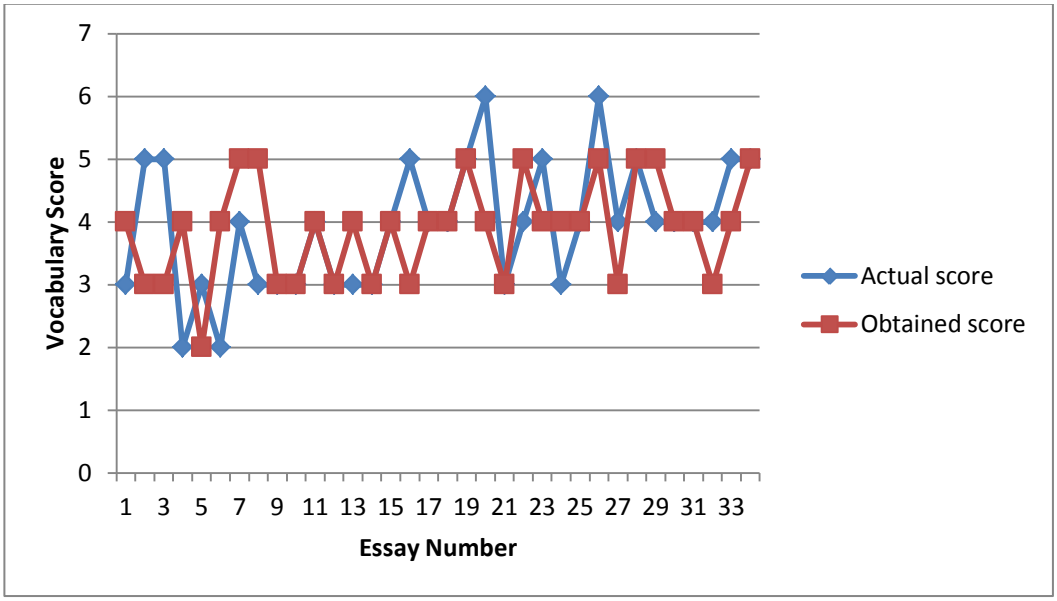



**Figure 7.8: Actual score versus obtained score for each essay**

### 7.4.3. Discussion of results

Reasons for adjacent agreement (where the algorithm assigned a lower score)

1. NAPLAN rubric awards a higher score to an essay if there are a higher number of precise words.

Examples of precise words are: predicament, unfathomably, perseverance, obese.

We tried to capture the number of precise words in an essay by counting the different words in each class: simple, common, difficult and challenging. Similarly, we tried to capture the quality of the precise words by including the average word length and the Flesch indices as features, but since NAPLAN lists are not comprehensive, we could not detect all the words in these classes. For example, we could not detect the word *constable* as difficult because it was not in the list. Also, we could not detect several other words, such as *persuading, discolouring, battlefield, tarnishing, mystified, armour, immaculate.* Hence, the lower score for some essays.

2. A higher score is awarded to the use of technical words such as *tranquiliser, exhaustion, fainted* (from the medical field), *confession, fingerprints, juvenile* (from law), *flashback and surround sound* (in electronics). Again, because of restrictive lists, a lower score is allocated to some essays.

3. The NAPLAN rubric assigns a higher score to an essay depending on the number of collocations. For example: 'things from the shops', 'on the spur of the moment', 'to have a holiday', 'walking other dogs', 'walk down the road' are correct collocations. Currently, there is no mechanism in the algorithm to identify collocations.

4. Furthermore, a higher score is awarded to the essay that uses metaphors. A metaphor is a phrase used to imply a subtle meaning but not the literal meaning. Some good examples of metaphors are: 'face stained with tears', 'sky was blood red', 'wind bit into his bones', 'ankle burned with pain'. Computers being machines, take the literal meaning into consideration.

Hence, detecting metaphors is difficult for a machine. Accordingly, we cannot detect metaphors at the moment.

5. A higher score is awarded to an essay that employs phrases depicting effective personification. Some examples of phrases portraying personification are: 'the door hung from its hinges as though it was trying to run away' and 'thoughts raced through Daniel's head'. Again, we cannot detect this at the moment because a computer takes literal meaning, being a machine.

6. A higher score is awarded to the use of colloquial speech for characters. Colloquial speech is also called informal language or ‘slang’. It is used in narrative writings to sketch an actor’s character for the audience. For example, words and phrases used in everyday life, such as: ‘Dunno’, ‘yep’, 'We're on it', 'Hold on man', 'No way man' and 'Hey babe'. We cannot detect this unless we use a slang dictionary.

7. A higher score is awarded to essays that use very poetic/ formal language to sketch an actor’s character. For example: ‘to what do I owe this honour?’. In the context of NAPLAN, younger writers i.e., primary school students of year 3 and 5, are still learning the basics of written language and will write stories with explicit description of story-line and conversation between characters of the story. On the other hand, older writers in high school would experiment with more subtle forms of writing like sarcasm, humour and metaphors, all of which is very difficult for the computer to grasp or detect. This is because a computer, being a machine, will take the literal meaning rather than the implicit meaning or the intended meaning. Also, we as humans sometimes have difficulty in identifying humor and satire in scripts.

8. A higher score is awarded to essays that use effective similes in the essay. For example: 'hair as a raven' and 'her skin like bark'. Currently, the algorithm cannot identify effective similes. This can be investigated as future work.

Reasons for adjacent agreement (where the algorithm assigned a higher score)

1. Where there was an incorrect usage of a collocation, the algorithm could not detect such errors, for example, 'helped the child up', 'trail left by the frightened horse at a joy' is an incorrect collocation. This can be overcome in the future by using a colocation detection technique, as explained in [1].

2. A thorough analysis of the errors leads us to suggest that the use of a comprehensive wordlist of the four classes will help us to grade the adjacent agreement essays more accurately.

In the next section, the main points of the chapter are presented as a recap and the chapter is concluded.

## 7.5. Conclusion

In this chapter, the methodology of the vocabulary module was described. This module is responsible for automatically assigning a grade for the vocabulary in an essay. Two separate algorithms form the core of this module. The algorithm for poor essays is used for assessing the vocabulary of an essay that has been detected as poor. The algorithm for good essays is used for evaluating the vocabulary of a good essay and the working of both the algorithms is described. The experimental simulation of the algorithms, the testing and the evaluation of their performance is described in detail. The perfect agreement and one-point adjacent agreement values for algorithm for poor essays were 65.2 and 34.8 respectively. For algorithm for good essays, the perfect agreement and one-point agreement values were 55.9 and 45.1 re-

spectively. The results of both algorithms are very impressive considering the fact that this is the first attempt at automating the vocabulary grading process according to NAPLAN rubric. Discussion of results and ways to improve performance is also presented.

In the next chapter, the overall framework of the methodology for automatically assessing the sentence structure in an essay is explained in detail. Also, the two algorithms, one for poor essays and the other for good essays are described as is their evaluation.

## 7.6. References


[1] C. Napoles and M. Dredze, "Learning simple Wikipedia: a cogitation in ascertaining abecedarian language," presented at the Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids, Los Angeles, California, 2010.

[2] C. K. Ogden, *Basic English: A General Introduction with Rules and Grammar*: Paul Treber & Co., Ltd., 1930.

[3] Voice of America. (2009, 5 August 2011). *Word Book (50th Anniversary ed.)*. Available: www.voaspecialenglish.com

[4] Cognitive Computation Group. (2010, 31 August, 2011). *Sentence Segmentation Tool*. Available: http://cogcomp.cs.illinois.edu/page/tools_view/2

[5] G. Williamson. (2009, 13 August 2011). *Type-Token Ratio.* Available: http://www.speech-therapy-information-and-resources.com/type-token-ratio.html

[6] R. Flesch, "A new readability yardstick," *Journal of Applied Psychology*, vol. 32, pp. 221-233, 1948.

[7] Y.-W. Lee, C. Gentile, and R. Kantor, "Toward Automated Multi-trait Scoring of Essays: Investigating Links among Holistic, Analytic, and Text Feature Scores," *Applied Linguistics*, vol. 31, pp. 391-417, 2010.

[8] L. S. Larkey, "Automatic essay grading using text categorization techniques," in *Proceedings of the 21st ACM/SIGIR (SIGIR-98)*, Melbourne, Australia, 1998, pp. 90-96.

# Chapter 8: Automated Marking of Sentence Structure

## 8.1. Introduction

In this chapter, the methodology and design of the sentence structure module is explained. The objective of this module is to perform automated scoring of the sentence structure criterion according to the NAPLAN rubric. To grade essays for sentence structure, it is important to develop two different approaches: one for grading poor essays and another for grading good essays. This is to ensure that we apply appropriate techniques to grade essays depending on the level of language skill and expertise demonstrated therein. Accordingly, in this chapter, a heuristics and rule-based algorithm for grading sentence structure in poor essays is outlined and expounded. Further, an artificial neural network based intelligent algorithm for grading sentence structure in good essays is elucidated. The working of the algorithm for poor essays is demonstrated with the help of sample essays. Additionally, for both algorithms, the simulation and the results obtained are presented. Finally, the performance of each algorithm is evaluated using a number of performance metrics widely used in the area of data mining and automated essay grading.

In the next section, the working of the sentence structure module is presented in detail.

## 8.2. Automated Scoring of Sentence Structure

The sentence structure module comprises of two main algorithms which perform the automated grading process. In this section, the methodologies of the algorithms for grading sentence structure in poor essays and the algorithm for grading sentence structure in good essays are explained in detail. The former is based on a set of heuristics and rules, whereas the latter uses certain NLP techniques and is based on artificial neural networks. During the preliminary part of both algorithms, the sentence structure capabilities of Microsoft Word 2007 are used to check if the sentence structure of a sentence is correct or not.

In the next section, the algorithm for grading sentence structure in poor essays is explained in detail.

## 8.3. Algorithm for Poor essays

As previously mentioned, poor essays contain mainly random typing errors, or are extremely poor in spelling and punctuation, or have less than 80 words. In fact, some essays have a copious amount of gibberish words to the extent that it is impossible to decipher the meaning of the words. Furthermore, these types of essays do not have proper sentence structures. Hence, a set of heuristics and rules based on the English language are used to develop the algorithm for grading sentence structure in poor essays.

The algorithm for poor essays is illustrated in Figure 8.1. In involves the following steps:

1. For the essay that is to be graded, calculate the percentage of content words (CW%). As discussed in chapter 7, section 7.3, content words are all the words in the essay that can be found in the *Basic Vocabulary* database. The value of CW% is obtained from the algorithm for grading vocabulary in poor essays.

   For the algorithm for grading sentence structure in poor essays, the empirically derived value of CW% is 25. So if the essay has a CW% value $\geq 25$, then it will be processed further by the system. Otherwise, the essay is flagged and a message "To be marked by human grader" is displayed.
2. The next step is to check if the total number of words in the essay (W) is 1. If so, it means that there are no sentences in the essay. This is because the shortest sentence that can be written in the English language contains at least two words (For example, "It's Friday" or "Go home"). The algorithm obtains the value of 'W' from the custom-developed program "SpellChecker", which provides the surface features of the essay in an XML file. According to the NAPLAN rubric, if the total number of words in the essay is 1, then a sentence structure score (SS) is assigned as '0' for such an essay. So, if W = 1, then the algorithms assigns SS a value of '0'. On the other hand, if the value of 'W' is more than 1, then it might mean that there is a sentence(s) in the essay.
3. The next step is to check if the number of sentences in the essay is 0 or 1. This can be determined by checking the 'Total number of sentences in essay' in the XML output obtained from the 'SpellChecker' program. Word counts the number of sentence end markers such as full stop or an exclamation mark or a question mark at the end of the sentences in the essay and gives the value as the total number of sentences in the essay. However, if the essay contains a sentence but does not contain any sentence end markers at the end of the sentence, then the number of sentence is shown as '0'. On

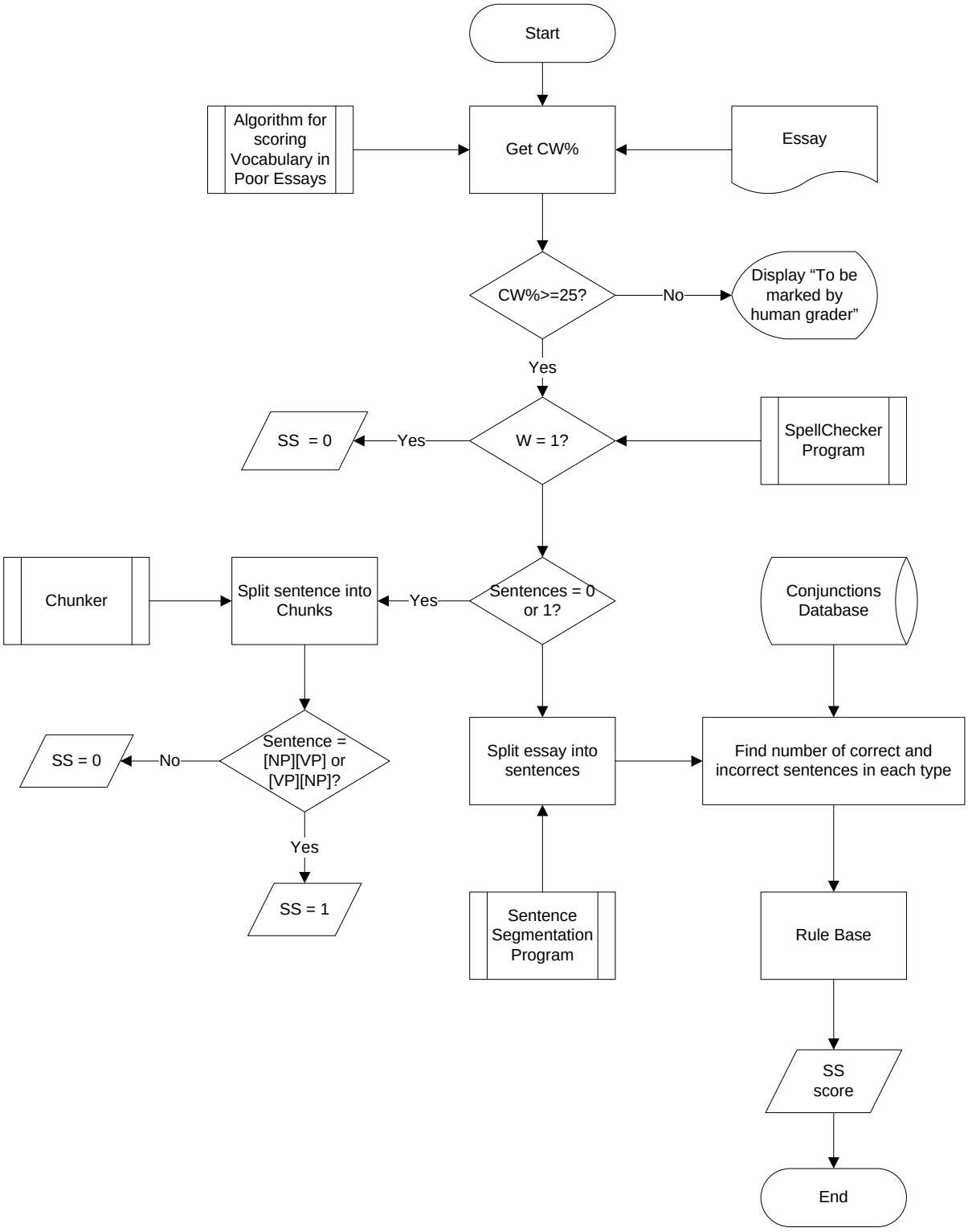



**Figure 8.1: Algorithm for grading Sentence Structure in poor essays.**

the other hand, if there is a sentence and a sentence end marker is detected, then the number of sentences in the essay is shown as '1'. Hence, if the condition "Sentence =

0 or 1" is satisfied, then we need to check if the sentence has a proper structure or not. According to Chomsky Phrase Structure Grammar, to have a proper structure, a sentence should have a noun phrase (NP) and a verb phrase (VP) [1]. In order to obtain the phrases in the sentence, we run the sentence through Illinois Chunker, which splits the sentence into phrases and denotes this by tags such as NP and VP. From the output obtained from the Chunker, we further check if it is of the form [NP][VP] or [VP][NP]. If yes, then SS is assigned as '1' else it is assigned as '0'.

4. If the total number of sentences in the essay is more than 1, then the next step is to split the essay into sentences. To do this, we use the 'Sentence Segmentation' code that is available with the Illinois suite of NLP tools and can be freely downloaded from [2]. The sentence segmentation tool reads plain text and rewrites it with one sentence per line, as the output.
5. Now for every sentence, we need to determine
    a. the type of sentence
    b. the number of correct and incorrect sentences in each type

According to the NAPLAN rubric, the type of sentence can be simple, compound or complex. In order to determine the type of a sentence, a "Conjunctions Database" has been compiled which contains the binding conjunctions and linking conjunctions, as given by NAPLAN. Binding conjunctions are the list of conjunctions that are used to join clauses in a complex sentence. Linking conjunctions are the conjunctions that are used to join clauses in a compound sentence. The complete list can be found in Appendix A.

To determine the type of a sentence, the algorithm first checks if it contains any binding conjunctions. If so, then the sentence is flagged as 'complex', else the algorithm further checks if it contains any of the linking conjunctions from the "conjunctions database". If so, then the sentence is flagged as 'compound', else the algorithm flags the sentence as 'simple'. In order

to determine if the sentence structure of the sentence is correct or incorrect, we check if the sentence is in <Grammar Errors>, in the XML file. If so, the sentence is tagged as incorrect, else the sentence is tagged as correct. At the end of this step, the algorithm determines:

- the type of sentences in the essay
- the number of correct 'simple' sentences = NumA
- the number of incorrect 'simple' sentences = NumWA
- the number of correct 'compound' sentences = NumB
- the number of incorrect 'compound' sentences = NumWB
- the number of correct 'complex' sentences = d

Further, the algorithm also computes the percentage of 'simple' sentences (SP) and the percentage of 'compound' sentences (CP) using the formulae in equations 8.1 and 8.2 respectively, as shown below.

$$SP = \left(\frac{NumA}{NumA+NumWA}\right) * 100 \qquad \text{.....Equation 8.1}$$

$$CP = \left(\frac{NumB}{NumB+NumWB}\right) * 100 \qquad \text{.....Equation 8.2}$$

While computing the above values, we ignore sentences which consist of only one word. For example, the sentence "No!". Although it might be treated as a sentence and given in the output of the 'Sentence Segmentation Process', we do not determine the type of such a sentence because it has no structure. Additionally, if we consider them and include them in the above computations, then the SS score result might not reflect the actual sentence structure.

6. The next step is to refer to the ‘Rule Base’ shown in figure 8.2, and assign the SS score accordingly.

```
If ((SP>=80)&&(CP>=80)&&(d>1)) then SS = 3 else

        If (((SP>=80)&&(CP>=80)) OR ((SP>=80)&&(d>1)) OR ((CP>=80)&&(d>1)) OR (d>1) OR
(SP>=80) OR (CP>=80))then SS = 2 else SS = 1
```

**Figure 8.2: Rule Base for assigning Sentence Structure score.**

The rule base contains the mathematical formulization of the NAPLAN rubric for the assignment of the SS value, depending on the number of correct and incorrect sentences of each type. Furthermore, an essay can contain all three types of sentences, that is, ‘simple’ AND ‘compound’ AND ‘complex’. Otherwise, the essay can also contain only ‘simple’ sentences (OR) only ‘compound’ sentences (OR) only ‘complex’ sentences (OR) a combination of these, such as ‘simple’ AND ‘compound’ (OR) ‘compound’ AND ‘complex’ (OR) ‘simple’ AND ‘complex’. With this in mind, the rule base was developed. In Figure 8.2, in order to assign an SS score of ‘3’, we check if the SP value is at least 80 AND the CP value is at least 80 AND the number of complex sentences is more than 1, in which case the SS is assigned a score of ‘3’. Otherwise, in order to assign a score of ‘2’, depending on the type of sentences present, we check if the SP value is at least 80 OR the CP value is at least 80 OR the number of complex sentences is more than 1 OR a combination of these conditions is satisfied. If not, then SS is assigned a score of ‘1’.

The algorithm for scoring sentence structure in poor essays is coded completely in the Java language. The algorithm was tested using sample essays and the outputs obtained from the program along with the results are detailed in the next section.

### 8.3.1. Sample Results

A sample output obtained for an essay, 'Dann_4.doc', is shown in Figure 8.3. The first step is to calculate CW%, which is determined to be about 17. Since CW% is less than 25, the essay

```
C:\Essay Grading\Poor Essays\Dann_4.doc
================================================================================
CW% = 17.391304347826086
================================================================================
"To be marked by human grader"
```

**Figure 8.3: Output obtained for sample essay Dann_4.doc.**

Consider another example of the sample output obtained from the program for another essay.

A sample output obtained for a poor essay is shown in Figure 8.4. The essay name is 'Bropho_2.doc', as mentioned in the first line in the output. The CW% of the essay is estimated at 50, which is more than 25, hence the value of 'W' is determined as the next step. Since the number of words in the essay is more than 1, the number of sentences is determined as the next step. Since 'Sentences = 0 or 1', the sentence is split into chunks using the Chunker. Hence, the 'Chunker Output' is given in Figure 8.4, showing the tags [PP][NP]. The condition 'Sentence = [NP][VP] or [VP][NP]' is not satisfied, hence the SS is assigned a score of '0', which is the same as that assigned by expert human markers.

```
C:\Essay Grading\Poor Essays\Bropho_2.doc
============================================================
CW% = 50,         W = 4
Number of sentences = 0 or 1
============================================================
Chunker Output
============================================================
[PP (IN FOUND)]
[PP (IN In)] [NP (JJ difficult) (NN situ)]
============================================================
SS Score is 0
```

**Figure 8.4: Output obtained for sample essay Bropho_2.doc.**

The sample output obtained for another poor essay, ‘Gandy_9.doc’, is shown in Figure 8.5. For the essay, the value of CW% is estimated at about 65, hence the essay is further processed. The total number of words in the essay (W) is determined to be 62 which is greater than 1. Hence, the next step is to determine the number of sentences in the essay, which is 9. Since ‘Sentence = 0 or 1’ is not satisfied, the next step is to split the essay into sentences by using the ‘Sentence Segmentation’ code.

```
C:\Essay Grading\Poor Essays\Gandy_9.doc
================================================================================
CW% = 65.625,               W = 62
Number of sentences = 9
 i found a lr at school .: Type is ‘Simple’
 i found a hat at school .: Type is ‘Simple’
 i found so mene mouse on the internet i like et i save it .: Type is ‘Complex’
 i found kindness for you today .: Type is ‘Simple’
 i found writing is yse .: Type is ‘Simple’
 i found a dog in the lake .: Type is ‘Simple’
 i found a choice to sit at school today .: Type is ‘Complex’
 i found a tens ball at school .: Type is ‘Simple’
 i found: Type is ‘Simple’
d = 2     NumA = 7          NumB = 0          NumWA = 0       NumWB = 0       CP = NaN          SP =
100.0
============================================================
Rules Trace
============================================================
---2---
SS Score is 2
```

**Figure 8.5: Output obtained for sample essay Gandy_9.doc.**

The output produced by the algorithm is of the form shown in Figure 8.5, where each sentence is displayed on a new line. Then the next step is to determine the type of sentences present and the number of correct and incorrect sentences in each type. The algorithm identifies the type of each sentence by using the logic given in the algorithm. Hence, the output shows each sentence (on the left-hand side of the colon), along with the type (on the right-hand side of the colon), as determined by the algorithm. Further, the number of sentences in each type are denoted by ‘NumA’ and ‘d’ as 7 and 2, respectively. But since there are no ‘compound’

sentences, the values of 'NumB', 'NumWB' and CP are 0 and NaN (which means it cannot be determined), respectively. The SP value is determined to be 100 because all the 'simple' sentences detected are not found in <Grammar Errors> in the XML file obtained from the 'SpellChecker' program.

The next part of the output in Figure 8.5 denotes the 'Rules Trace' which gives details of the rule that is satisfied by the program. Internally, the rules are numbered by the Java program and the numbers are used to denote the rule that is satisfied. In the 'Rules Trace', number '2' denotes that the second rule 'If (((SP>=80) && (CP>=80)) OR ((SP>=80) && (d>1)) OR ((CP>=80) && (d>1)) OR (d>1) OR (SP>=80) OR (CP>=80)) then SS = 2', is satisfied. Finally, the SS value is displayed as '2', which is the same as that assigned by expert human markers.

### 8.3.2. Performance Evaluation

Testing was carried out for the essays from the 'Poor Essays' dataset. The results obtained during the testing phase are presented in Table 8.1. An essay dataset of 135 poor essays was obtained during the filter process, explained earlier in Chapter 4. Of the 135 essays, 18 essays did not satisfy the condition 'CW% ≥ 25' of the algorithm, hence they were flagged 'To be marked by human grader'. The remaining 117 essays in the poor essay dataset were used to test the algorithm.

Since the algorithm for scoring sentence structure in poor essays scores and rates an essay into one of the score categories from 0 to 3, we use the widely recognised rater agreement metrics as performance metrics, as mentioned previously in Chapter 6, to report the performance of the algorithm.

We run the essays through the program for scoring sentence structure in poor essays. Our aim is to match the human-assigned scores in order to validate our system performance. We com-

pare our system assigned score with the human-assigned score in order to achieve this. Perfect agreement, one category adjacent agreement, two category adjacent agreement and finally perfect + adjacent agreement are the measures used to present the results.

**Table 8.1: Agreement values for the algorithm for poor essays.**

| | Number of essays (of n = 117) | Percentage |
|---|---|---|
| Perfect Agreement | 59 | 50% |
| One-point Adjacent Agreement | 50 | 43% |
| Perfect + Adjacent agreement | | 93% |
| Two-point Adjacent Agreement | 8 | 7% |
| Total | | 100% |

Of the 117 essays in the dataset, a perfect agreement score was obtained for 59 essays, one-point adjacent agreement was obtained for 50 essays and two-point adjacent agreement was obtained for 8 essays. As a result, it can be concluded that the algorithm achieved a perfect agreement of about 50%, a one category disagreement of about 43% and a two-point adjacent agreement score of about 7%, which is quite good. There were no essays which received a non-adjacent agreement score. Moreover, perfect + adjacent agreement is 93% which is exceptionally good. The results in Table 8.1 demonstrate that our system can grade a poor essay with 93% accuracy within one score point. Complete results can be found in Appendix E.

### 8.3.3. Discussion of results

Our algorithm is mainly governed by the sentence end markers as indicated by students. This is the major source of error due to the fact that computers cannot recognise the end of a sentence if there is no sentence end marker provided. Hence, in cases where students used more

sentence markers than required, our algorithm gave a higher score whereas in cases where they used less sentence markers than required, our algorithm gave a lower score.

In the next section, the working of the algorithm for good essays is explained in detail.

## 8.4. Algorithm for Good essays

Figure 8.6 illustrates the algorithm for good essays. This algorithm involves the extraction of certain features related to sentence structure. Then, these features are input to a neural network model to simulate and obtain an output as the sentence structure score for the essay.

The various steps involved are:

1. Split the essay into sentences. We use the Stanford Sentence Segmentation tool to do this. It takes the text of the essay as input and prints each sentence on a new line as the output.
2. Determine variety in sentence length. To determine if there is a variety of sentence lengths in the essay, we first calculate the length of each sentence obtained in step 1. The number of words in the sentence is the length of the sentence. Then, depending on sentence length, each sentence is assigned a value in a particular range, as shown in Figure 8.7. At the end of this step, if the range-list has entries in at least 3 ranges, then the variety is set to ‘True’ else it is set to ‘False’.

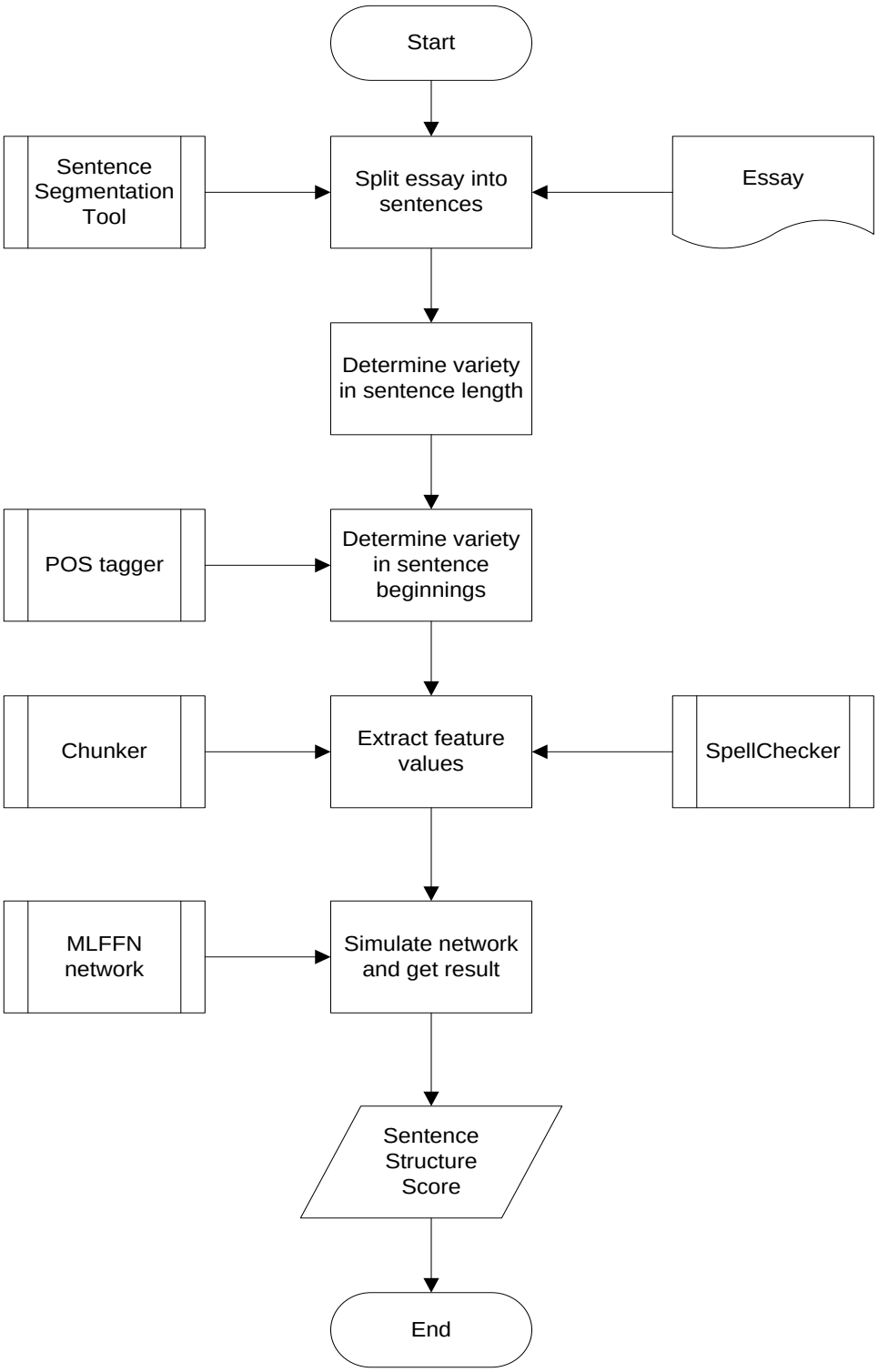



**Figure 8.6: Algorithm for grading sentence structure in good essays.**

*For every sentence in essay, do*
*Calculate sentence length = total words in sentence*
*Allot sentence in the range-list 1-10, 11-20, 21-30, 31-40, 41-50*
*End*
*if the range-list has entries in ATLEAST 3 ranges, then VarietyLen = True else False.*

**Figure 8.7: Determining variety in sentence length**

3. Determine variety in sentence beginnings. To determine if there is a variety of sentence beginnings in the essay, we first run each sentence obtained in step 1 through the Stanford POS tagger. It takes a sentence as input, attaches the POS tag to each word and displays the words and their tags as output. From this output, we obtain the POS tag of only the first word in the sentence, as shown in Figure 8.8. This process is carried out for each sentence in the essay. Then, if the frequency of a particular POS tag is at least 50% then VarietyBegin is set to 'True', else it is set to 'False'.

*For every sentence in essay, do*
*Get POS tag of first word*
*End*
*Frequency of most-repeated POS tag = (number of times the POS tag is found/total number of sentences)*100*
*if frequency of a particular POS tag ≥ 50% then VarietyBegin = False else True.*

**Figure 8.8: Determining variety in sentence beginnings**

4. Extract feature values. In this step, we determine the values of various features related to the syntactic structures of the sentences, sentence structure and fluency, as highlighted in [3] which reports on the various features that influence sentence fluency. Some of the features that we employ for essay grade prediction are average sentence length, number of subordinating conjunctions (SBAR count including SBARQ), number of noun phrases (NPs, including WHNPs), verb phrases (VPs), prepositional phrases (PPs including WHPPs), adjectival phrases (ADJPs), adverbial phrases

(ADVPs including WHADVPs) and embedded clauses. Their findings suggest that as the length of the noun phrase increases, the complexity of the sentence increases.

We input each sentence obtained in step 1 to the Illinois Chunker. It takes a sentence as input, assigns tags to syntactically-related phrases in the sentence and then prints the sentence and tags as output. From this output, we obtain the following features:

i. the total number of SBAR and SBARQ tags
ii. the total number of NPs
iii. the total number of VPs
iv. the total number of PPs
v. the total number of ADJPs
vi. the total number of ADVPs
vii. the length of longest NP. The length of NP=number of words in the NP (but we do not count the punctuation marks tagged as NP). Suppose in the first sentence NP length = 3, then check the second sentence. If NP length>3, then update. Then check the third sentence. If NP length>current value, then update again and so on, until the end of the essay.
viii. Ratio1 = (2)/(3) where 2, 3 are the number of NPs and number of VPs, respectively
ix. Ratio2 = (4)/(2) where 4, 2 are the number of PPs and the number of NPs, respectively.
x. Ratio3 = (5)/(2) where 5, 2 are the number of ADJPs and the number of NPs respectively.

From the SpellChecker program, we obtain values for the total number of words in the essay, the total number of sentences in the essay and the ‘normalised grammar error by sentence’ value.

5. In this step, we use MATLAB to create a neural network and feed input to the network in order to determine the essay grade as the output. This is explained in detail in section 8.3.4.
6. In this step, the sentence structure score obtained from the neural network model is displayed.

### 8.4.1. Neural Network Design and Calibration

To choose the optimal neural network, networks of various configurations are to be developed and simulated using the dataset. Using the MATLAB GUI tool for neural networks, 'nntool', we created multi-layer feed forward neural (MLFFN) networks of various configurations, as shown in Table 8.2. For each network, training was performed using the training set. The network was calibrated by retraining over a number of iterations until it produced a low MSE value and until the network performed satisfactorily. Then, using the testing set, the network was simulated and the results are reported.

**Table 8.2: Details of neural network architecture designed for good essays.**

| | |
|---|---|
| Type of neural network | Multi-Layer Feed Forward Neural Network (MLFFN) |
| Number of hidden layers | 2 |
| Number of neurons in hidden layer | 55, 60, 65 |
| Training algorithm | 'trainlm' |
| Learning Algorithm | Back propagation |
| Learning functions | 'Tansig', 'tansig' |

We configured three different MLFFN networks with 2 layers in the hidden layer for each network. The first network had 55 neurons in the hidden layer. Then, we increased the num-

ber of neurons by 5, so the second network had 60 neurons in the hidden layer. We performed the training and calibration of this network in the same way as we did for the first network. Then, we increased the number of neurons in the hidden layer by 5. Hence, the third network had 65 neurons. For each network, the training algorithm is 'trainlm' and the learning algorithm is 'back propagation'. The learning functions for both hidden layer 1 and hidden layer 2 is 'tansig' because this is a pattern classification task.

The complete dataset consisting of 172 good essays is divided using the stratification technique as explained previously. Hence, using the above technique, the training set consists of 138 essays and the testing set consists of 34 essays.

Using the GUI shown in Figure 7.5 in Chapter 7, the data for the training phase is chosen. The inputs and targets are specified and then the network is trained. The outputs of this phase are stored as training results.

The results obtained during the training phase of each network were recorded and used to calculate the root mean square error (RMSE) was calculated using formula I.

$$\text{RMSE} = \sqrt{\frac{\sum \left(f(x_i) - y_i\right)^2}{n}} \quad \text{.....I}$$

where $f(x_i)$ = Target value of $i^{th}$ essay,

$y_i$ = Result obtained for $i^{th}$ essay,

n = Total number of essays in dataset.

In Table 8.3, the RMSE values for the training phase are shown for different configurations of the network. The optimal performing network is the one which provided the least MSE and

hence, it was chosen as the final model. The results are shown in table 8.3 and the RMSE of the various models is shown in Figure 8.9.

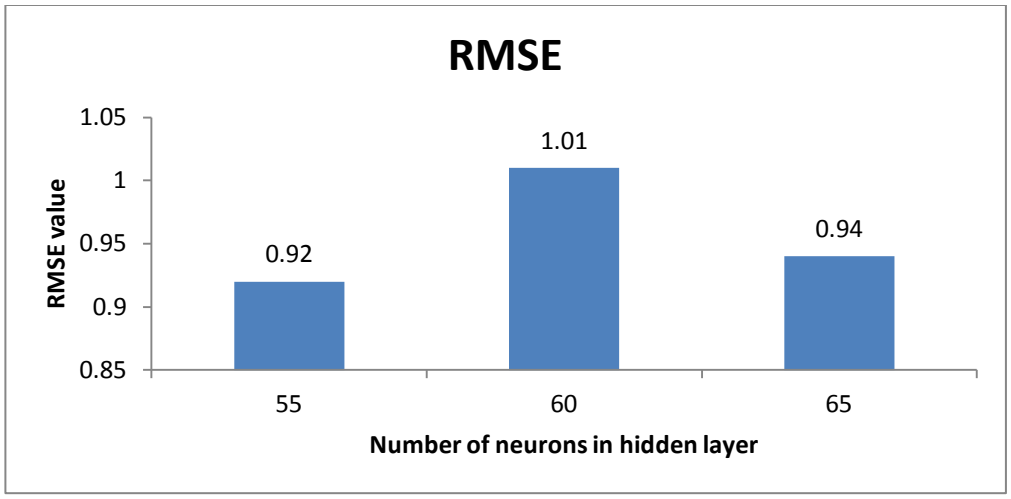



**Figure 8.9: RMSE values for neural network models**

The network with neurons = 55 is chosen as the final model because it produced the lowest mse. Then this network is simulated using the testing set.

### 8.4.2. Performance Evaluation

Using the GUI shown in figure 7.7 in chapter 7, the data for the testing phase is chosen. The inputs and targets are specified for simulation and then the network is simulated. The outputs of this phase are stored as simulation results.

The neural network model that produced RMSE of 0.92 during the training phase provided RMSE value of 1.09 during the simulation phase. Additionally, using our previous performance metrics of agreement rates, the values for exact agreement, adjacent agreement rate and perfect+adjacent agreement rate are determined and the results are shown in Table 8.3 below. Exact agreement was obtained for 15 of 34 essays which is 44.1% , one-point adjacent agreement was obtained for 12 essays which is 35.3%, two-point adjacent agreement was obtained for 7 essays which is 20.6% and finally, the perfect+adjacent agreement is 79.4%.

Hence, we can conclude that our system can grade sentence structure in good essays with an overall accuracy of nearly 80%, which is quite good.

**Table 8.3: Agreement rates obtained from the neural network model.**

| | Number of essays (of n = 34) | Percentage |
|---|---|---|
| Perfect Agreement | 15 | 44.1% |
| One-point Adjacent Agreement | 12 | 35.3% |
| Perfect+Adjacent Agreement | | 79.4% |
| Two-point Adjacent Agreement | 7 | 20.6% |
| Total | | 100 |

The detailed results are shown in Figure 8.10. The x-axis denotes the essay number and the y-axis denotes the output score range, which is from 2 to 6 for good essays. The blue markers show the target output for the essay and the red markers show the result obtained from our system. For essays where only one marker is present, this means that there is perfect agreement between targets and the result. For all other essays, the discrepancy between the target and result value for the essay is shown as 1 or 2. Hence, from the figure, it is evident that

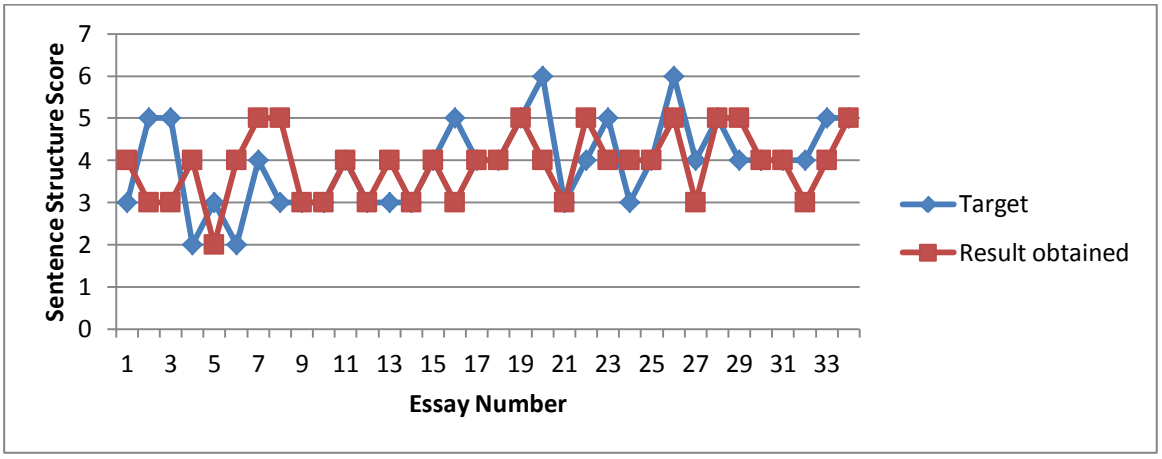



**Figure 8.10: Target values versus results obtained for each essay**

there is perfect agreement for 15 essays, a one-point adjacent agreement for 12 essays and two-point adjacent agreement for 7 essays.

Therefore, there is a discrepancy between the targets and results obtained for 19 essays in the dataset. The possible reasons for the discrepancies are investigated and presented in the next section along with ways to overcome or avoid the discrepancies in future.

### 8.4.3. Discussion of results

The reasons some essays received a one-score point or two-score point discrepancy from the actual score are as follows.

1. We used the 'normalised grammar error by sentence' value for the essay as an indicator of the errors in the sentence structure. However, Microsoft Word 2007 includes some kinds of punctuation errors in grammar errors. Due to this, for some essays, the grammar error value was not the exact indicator of the errors in sentence structure. This can be corrected in the future by using the appropriate settings in Word 2007.
2. Some essays contained sentences which were syntactically correct but not very meaningful. Human markers scored such essays lower whereas since we did not use a semantic parser to analyse the meaningfulness of sentences, our system scored such essays slightly higher. However, these kinds of essays are not found very often.

In the next section, a recap of the main points of this chapter is presented and the chapter is concluded.

## 8.5. Conclusion

In this chapter, the overall methodology of the sentence structure module was provided. The two innovative algorithms that constitute the module were presented and explained, these being an algorithm for grading sentence structure in poor essays and an algorithm for grading sentence structure in good essays. The logic for poor essays is based on heuristics and a rule-based approach whereas the logic for good essays is based on neural network modelling. The details of both algorithms and their working were explained. Using sample essays, the testing was carried out for the algorithm for poor essays and the results were provided and discussed. Then, using the testing results, the performance of the algorithm was evaluated using the standard performance metrics of exact, adjacent and perfect+adjacent agreement rates.

For the algorithm for good essays, testing was carried out using the testing set and the results were explained. The performance is evaluated first by using standard performance metrics of exact, adjacent and perfect+adjacent agreement rates and then, by null hypothesis testing. The perfect+adjacent agreement rate obtained from the results was nearly 80%. Considering the fact that both the algorithms that comprise the sentence structure module are a first of their kind, their performance is considered to be very successful. Finally in this chapter, discussion was undertaken to analyse the discrepancy in the obtained score from the algorithms and the actual score. Thereafter, several suggestions were made to improve the performance of both algorithms.

## 8.6. References


[1] N. Chomsky, "Three models for the description of language," *Information Theory, IRE Transactions on*, vol. 2, pp. 113-124, 1956.

[2] Cognitive Computation Group. (2010, 31 August, 2011). *Sentence Segmentation Tool*. Available: http://cogcomp.cs.illinois.edu/page/tools_view/2

[3] A. Nenkova, J. Chae, A. Louis, and E. Pitler, "Structural features for predicting the linguistic quality of text: applications to machine translation, automatic summarization and human-authored text," in *Empirical methods in natural language generation*, K. Emiel, Mari, and T. t, Eds., ed: Springer-Verlag, 2010, pp. 222-241.

# Chapter 9: Conclusion

## 9.1. Introduction

In this chapter, a summary of the main points of the thesis is presented and future direction is highlighted. Firstly, the research aims and objectives of this thesis are revisited along with the discussion as to how far they have been achieved in the thesis. A summary of the results obtained in this research is mentioned. In light of the achievements, the contribution of this thesis to the field of automated essay grading is emphasized. Then future work is proposed in order to propel further, the work done by this thesis. While discussing future work, ways to achieve performance of the various algorithms presented in this thesis have also been proposed. Finally, the limitations of this thesis are listed and the chapter is concluded.

In the next section, a recapitulation of the research aims and objectives of this thesis is presented and the extent of their achievement is discussed.

## 9.2. Recapitulation of research aims

As mentioned earlier, in order to address the problem, the following research aims and objectives have been formulated:

1. To develop an AEG system which is capable of grading essays in the English language and can handle improperly constructed responses.
2. To develop modules for the analytic scoring of narrative essays. The modules should be able to model both the linear and non-linear relationships between the essay features and its grade, should be computationally non-intensive and should be able to be trained and calibrated using a relatively small dataset. This research issue is further sub-divided into three research issues as follows:
    i. Develop a module for grading the spelling criterion, according to the NAPLAN rubric.
    ii. Develop a module for grading the vocabulary criterion, according to the guidelines stated by the NAPLAN rubric.
    iii. Develop a module for grading the sentence structure criterion, according to the NAPLAN rubric.
3. To verify and validate the methodologies developed in each of the research aims above.

In the following part of this section, it is discussed how far the research aims and objectives have been achieved in this thesis.

### 9.2.1. AEG system capable of handling improper responses is developed

The AEG system developed in this thesis is capable of grading essays written in English language. This is achieved by using rules and heuristics based on English language while designing the algorithms; using databases containing word lists and specifics related to English language and finally training the system using an essay dataset written in English. In order to handle improperly constructed responses, the filter process was proposed and implemented. The main purpose of the filter process was to separate anomalous and poor essays from the dataset. The different types of anomalous essays that can be detected are blank responses, responses which contain pictures instead of essay, responses which contain pictures and/or a few words, responses which contain the question prompt either partly or completely and finally, responses which are completely in upper case letters. The different types of poor essays that can be detected are essays which are too poor in spelling and/or grammar; and essays which are too short. As a result of the filter process, all these essays are separated from the essay dataset that goes into the grading process.

### 9.2.2. Modules for analytic scoring of narrative essays are developed

The grading process in the AEG system is performed by three different modules that constitute the grading process - A module each for grading the criteria spelling, vocabulary and sentence structure. The modules are capable of processing the essay according to the NAPLAN rubric and assigning an analytic score for narrative essays. The vocabulary and sentence structure modules model both the linear and the non-linear relationships between the essay features and its grade by using neural networks based models for grading good essays. Multilayer feed forward neural network models with back propagation algorithm as the training algorithm are used in the modules. Additionally, all three modules are computationally non-intensive and make use of available resources. Two freely available resources such as the POS tagger and chunker are used in the vocabulary and sentence structure modules respec-

tively. Moreover, databases are compiled and accessed locally by the programs on a local machine. Algorithms are designed in such a way that computations are non-intensive thereby the system is able to grade an essay in a matter of few seconds about 4 seconds per essay. The size of the essay dataset used to train and calibrate the system is 172 essays as mentioned previously. This is a relatively small requirement considering the datasets required by other AEG systems.

The spelling module can assess spelling in the essay in three steps. Firstly, the correct and incorrectly spelled words in the essay are identified. Then each word is classified into one of the four classes – simple, common, difficult or challenging by the Word Classification algorithm. Finally, the percentage of each class of words is calculated and a spelling mark is assigned to the score by the Spelling Mark algorithm.

The vocabulary module consists of two algorithms – one for grading poor essays and another for grading good essays. The algorithm for poor essays identifies the content words in the essay and depending on the percentage of them, assigns a score. On the other hand, the algorithm for good essays is more complex. Firstly, it extracts from the essay, various features related to vocabulary. Then a neural network model is calibrated by feeding the extracted features as inputs. This model produces the final score for vocabulary.

The sentence structure module consists of two algorithms – one for grading poor essays and another for grading good essays. The algorithm for poor essays firstly identifies the number of sentences in the essay. Then each sentence is analysed for correctness and later classified into one of three types – simple, compound or complex. The number of correct and incorrect sentences in each type is identified and so are their percentages. Then using mathematical formulation of the NAPLAN rubric, the final score is assigned by the algorithm. On the other hand, the algorithm for good essays involves the extraction of features related to sentence

structure from the essay. Then, these features are input to a neural network model to simulate and obtain an output as the sentence structure score for the essay.

### 9.2.3. Proposed methodologies are validated

The verification and validation of all the methodologies is undertaken using a real world essay dataset provided by WADET. The filter process is tested using a dataset of 308 essays. The filter process performs with a precision, recall and F-measure of 0.92, 0.96 and 0.97 respectively, which is very promising. The Word Classification algorithm was tested with a dataset of 700 words. The results obtained were the Word Classification Algorithm performs with a precision, recall and F-measure of 0.73, 0.71 and 0.69, respectively, which is quite promising. The Spelling mark algorithm was tested with a selection of 14 essays. The Spelling Mark algorithm performs with a perfect agreement rate of about 57%, a one-point adjacent agreement rate of 36% and a two-point adjacent agreement rate of 7%. Although it might seem that the results are not very good, the metric of perfect + adjacent is 92.9% which is exceptionally good for the first attempt at scoring spelling according to NAPLAN rubric.

The verification of the algorithm for poor essays is performed with a dataset of 135 essays. The performance of the algorithm was percentage of perfect agreement and one-point adjacent agreement values 65.2% and 34.8% respectively. During the validation of algorithm for good essays, a dataset of 34 essays was used. The perfect agreement obtained was 55.9% and the one-point adjacent agreement obtained was 44.1%.

For the verification of algorithm for poor essays 117 essays were used. The algorithm achieved a perfect agreement of about 50%, a one-point adjacent agreement of about 43% and a two-point adjacent agreement score of about 7%. The metric of perfect + adjacent agreement is 93% which is exceptionally good. The validation of algorithm for good essays was performed using 34 essays. Exact agreement obtained was 44.1%, one-point adjacent agree-

ment obtained was 35.3%, two-point adjacent agreement obtained was 20.6% and finally, the perfect+adjacent agreement is 79.4%. Hence, we can conclude that our system can grade sentence structure in good essays with an overall accuracy of nearly 80%, which is quite good.

In the next section, the various contributions of this thesis to the field of AEG are emphasized.

## 9.3. Contributions of this thesis

In this section, the significance and contributions of this thesis are highlighted. By using the filter process, almost all essays with a potential score of $\leq 15$ are detected successfully. Moreover, if a student tries to trick the AEG system by writing a lot of random words in order to acquire a high vocabulary score, the essay will be detected as a poor essay because of many sentence level mistakes. The sentence level mistakes will be because of incorrect sentence formations. The significance of Word Classification algorithm is that all over Australia, the class of a word would be the same thereby uniform scoring of the spelling criterion can be achieved. The importance of the spelling mark algorithm is that it will reduce bias in marking spelling because it actually counts the number of words in each class for every essay rather than making a guess like most human markers currently do.

The prominence of the algorithm for poor essays is that with such limited use of resources, it delivers more than 80% accuracy. The work-around developed is a very innovative way of scoring vocabulary. Instead of identifying the parts of speech of the words used in the essay,

we merely identify whether the words can be found in the database. The database actually contains a collection of words of various parts of speech belonging to basic vocabulary, as agreed by the experts. Compiling the *Basic Vocabulary* database for the purpose of scoring vocabulary is a significant contribution. The algorithm for assessing vocabulary in good essays is a novel and innovative algorithm that employs artificial intelligence based techniques such as the neural networks. If a student uses more synonyms in the essay, then the algorithm detects them and assigns a higher score by taking into account the higher type-token value in the respective class of words.

The algorithm for grading sentence structure in poor essays is a first of its kind. Keeping in mind the rudimentary and non-conventional nature of poor essays, the performance of the algorithm is very promising. Moreover, the algorithm for grading sentence structure in poor essays can detect extremely gibberish essays and highlight them as “To be marked by human grader”. Hence, intervention required from the human marker is significantly reduced as a result of using this algorithm. On the other hand, the algorithm for scoring sentence structure in good essays is a novel and innovative approach which employs neural networks. The algorithm used only shallow parsing methods and yet produced satisfactory results in assigning a score according to NAPLAN rubric, which has not been done previously.

In the next section, future work related to this thesis is discussed.

## 9.4. Future Work

In this section, the direction of future work related to each module of the AEG system and the system as a whole is highlighted in detail.

In the filter process, to further improve the performance of the noise detection methodology, the detection of syntactic and semantic gibberish can be done. As a result, essays which are syntactically correct but are meaningless can also be detected during the filter process. This will also prevent students trying to trick the system by writing lengthy but meaningless essays. Furthermore, the filter process cannot detect off-topic essays and essays copied from other students' work yet. This can be done as part of future work.

In the spelling module, the results obtained by the Word Classification Program can be a used to create a comprehensive NAPLAN database. To achieve this, human expert markers at WADET can carefully analyse the classification of words obtained from the algorithm and approve their classification as correct. Each word and its class that has been verified to be correct can be stored in the NAPLAN database for future reference. In this way, a comprehensive database can be created, which will grow over time. This database in turn can be used by the Spelling Mark algorithm to enable the automated scoring of spelling of essays which will improve the performance of the spelling mark algorithm as well.

In the vocabulary module, to improve the performance of the algorithm for poor essays in future, the first suggestions for incorrectly spelt words given by Microsoft Word can be obtained and included in the list *Unique*. Moreover, in order to improve the vocabulary calculation, it would be helpful to include the number of words in each class such as 'simple', 'common', 'difficult' and 'challenging'. This is because we are interested in the number of

precise words used (the more precise the word, the higher its class. For example: ‘obese’ is classified as ‘challenging’) and the vocabulary is scored higher for essays which use technical words, such as ‘resuscitated’, again a ‘challenging’ word. Finally, vocabulary calculation can be improved by the use of a collocation detection tool, as elucidated in the work by [1]**.** Collocations are commonly used phrases of words, such as "strong tea", "throwing a party" and “ride a bike”. This might improve performance even more. On the other hand, to improve the performance of algorithm for good essays, the comprehensive database developed by human expert markers (as mentioned above) can be used. This will enable the algorithm to identify classes of words which are not in the NAPLAN word classification database and will significantly boost the performance of the algorithm.

In the sentence structure module, the performance of the algorithm for poor essays could be improved significantly if the AEG system is able to detect the ending of a sentence, even in the absence of sentence end markers. Hence, in future effort is required to develop such a mechanism. In the algorithm for good essays a syntactic parser is used to analyse the sentence structure. The syntactic parser is designed to tag the syntactic structure of sentences, irrespective of whether the sentence is semantically correct or not. In future, to check the meaningfulness of sentences, semantic parsing can be used. A semantic parser is equipped with words, their respective concepts and rules to detect grammar errors, such as incorrect phrases, in sentences. Furthermore, in accordance with the work in [2], a complete structure-based grammar can be developed to detect basic and sophisticated structures of compound and complex sentences, and incorporated into the algorithm for the purpose of NAPLAN marking.

In the AEG system as a whole, there is no feature of plagiarism detection yet. This will help to identify essays which are not original contributions of the students and it can be added in future. Furthermore, the system has not yet been tested on essays of other genres. As part of future work, the algorithms can be modified for analytic scoring of other rubrics and used for

automated grading of essays. Additionally, the system can be modified for scoring essays of other genres as well. Furthermore, the framework can be implemented using fuzzy logic for dealing with vague human concepts such as treating untrained student's misspelling. The AEG system can be further enhanced by incorporating the implementation of marking of various other features of NAPLAN as shown in [3].

In the next section, a summary of the main points of this chapter are presented and the chapter is concluded.

## 9.5. Conclusion

In this chapter, firstly, the research aims and objectives of this thesis were mentioned and the results of our research are presented to assess as to how far they have been achieved in this thesis. Then, the important contributions of this thesis are listed. Finally, conclusions are drawn from the work done and steps for future work are highlighted.

## 9.6. References

[1] Y. Futagi, "The effects of learner errors on the development of a collocation detection tool," presented at the Fourth workshop on Analytics for noisy unstructured text data, Toronto, ON, Canada, 2010.

[2] U. C. Jaiswal, R. Kumar, and S. Chandra, "A Structure Based Computer Grammar to Understand Compound-Complex, Multiple-Compound and Multiple-Complex English Sentences," in *International Conference on Advances in Computing, Control, & Telecommunication Technologies*, 2009, pp. 746-751.

[3] H. W. Lam, T. Dillon, and E. Chang, "Determining Writing Genre: Towards a Rubric-based Approach to Automated Essay Grading," presented at the IEEE International Conference on Advanced Information Networking and Applications (AINA), Singapore, 2011.

# Appendix A

***Linking Conjunctions List(link two independent clauses-Compound)***

A sentence is called a 'Compound' sentence if it has two independent clauses, linked with a conjunction. The different types of linking conjunctions are:

i. Temporal sequence-time
    1. then
    2. and then
ii. Show cause and effect
    1. and so
    2. so
    3. and thus
iii. Adding
    1. and
    2. not only…but also
iv. Comparing/Contrasting
    1. but/but not
    2. or/either..or
    3. neither…nor
    4. yet
    5. except
    6. else

***Binding Conjunctions List(add a dependent clause-Complex)***

A sentence is called a 'Complex' sentence if it has a dependent clause , binded to the independent clause with the help of a conjunction. The different types of binding conjunctions are:

i. Timing (When?)
    1. after
    2. before
    3. as
    4. while
    5. when
    6. whenever
    7. just as
    8. as soon as
    9. until
    10. now that
    11. as long as
    12. since
    13. every time
ii. Show cause and effect (Why?)
    1. because

2. so that
3. in order that/in order to/to
4. since
5. if
6. even if
7. in case
8. although
9. unless
10. inspite of
11. despite
12. as long as
13. on condition that

iii. Adding (And what else?)

1. as well as
2. besides
3. along with
4. apart from
5. on top of
6. in addition to

iv. Manner (How?)

1. by
2. through

v. Comparing/Contrasting (Compared with what?)

1. like/just like/like when
2. while
3. whilst
4. as
5. as if
6. as though
7. whereas
8. although
9. except that
10. compared with
11. rather than
12. instead of
13. the way that

# Appendix B

## Word Classification Algorithm Results for 700 words

| Simple | Common | Difficult | Challenging |
|---|---|---|---|
| a | able | ancient | accelerating |
| add | above | accurate | accidentally |
| ago | again | anxiously | vegetarianism |
| all | air | excitedly | accumulate |
| am | along | pleasant | acquainted |
| an | always | structures | acquire |
| and | anybody | whispered | adrenaline |
| are | aren't | surveyed | aisle |
| as | asleep | blemish | appearance |
| at | backyard | existence | appreciated |
| ate | beach | hesitation | awkwardly |
| away | behave | accepted | baulk |
| bad | behind | approached | beige |
| bark | between | practice | belligerence |
| bee | bleed | eventually | benefited |
| bell | blind | wondered | annihilate |
| best | bought | similar | brevity |
| big | carries | horizons | brilliance |
| bin | chain | whatsoever | appropriate |
| bird | class | detectives | buoy |
| blow | color | instinctively | camouflage |
| book | trying | mourning | carcasses |
| box | cracked | aliens | climatic |
| had | dead | popular | colloquial |
| but | didn't | injured | colossal |
| by | discuss | weightless | column |
| can | doesn't | cautious | competence |
| car | don't | challenged | complementary |
| cheek | draw | determined | complimentary |
| clap | during | impossible | conscience |
| cow | morning | injuries | conscious |
| cup | everyone | alien | consequently |
| day | everywhere | embedded | courageous |
| deep | explain | vendetta | debris |
| did | fighting | attention | decomposed |
| dog | finally | autograph | deficient |
| doll | flight | persevered | definitely |
| dot | followed | beautiful | dependency |
| drag | found | bough | desiccate |
| dress | fruit | propelled | desperate |
| drip | goodness | abandoned | desperation |
| drop | green | acknowledge | dominant |

| | | | |
|---|---|---|---|
| drum | ground | certain | **draught** |
| eat | hair | chocolate | **effervescent** |
| egg | hatch | college | **efficient** |
| end | heaps | competition | **embarrassed** |
| fat | hearing | consider | **euphoric** |
| **ever** | **holidays** | **considerate** | exaggerate |
| feel | board | convince | exhilarating |
| feet | huge | crevice | **explanatory** |
| fell | hunted | crystal | fascinating |
| fill | **important** | dangerous | facilities |
| fit | inside | delicious | **gauge** |
| **five** | jacket | advantage | inconsequential |
| food | kitten | agencies | **grandeur** |
| for | **knee** | agreeable | **guaranteed** |
| **four** | large | allergic | **guillotine** |
| from | laying | annual | **gynaecology** |
| fun | leaving | **dye** | haemoglobin |
| get | letter | decision | hallucinate |
| go | little | attempt | **hesitance** |
| **going** | live | endangered | humanitarian |
| good | loud | enjoyable | incandescent |
| got | **magazine** | episode | incompetent |
| grass | many | attractive | inconsolable |
| hand | medals | auction | incorporate |
| hard | menu | extremely | indecipherable |
| has | **migrate** | features | insanity |
| hat | Monday | February | interrogate |
| **have** | moral | fiction | **intriguing** |
| he | movie | awesome | **iridescent** |
| help | naughty | behaviours | **irresponsible** |
| her | necklace | furniture | **judicial** |
| him | nephew | benefit | kaleidoscope |
| hot | noisy | gigantic | **kayaking** |
| how | **octopus** | **goblet** | **lacerate** |
| I | once | graphics | **lieutenant** |
| if | onion | hammock | **liquefy** |
| in | outdoors | boulder | **longevity** |
| **into** | outside | **hesitated** | **luminescent** |
| is | panic | brethren | **magnificent** |
| it | **paw** | hopefully | malaria |
| **just** | **picture** | hygiene | **mandible** |
| keep | planet | **imaginative** | **manoeuvre** |
| kid | platform | impressed | mathematician |
| land | power | **information** | **mediaeval** |
| left | **princess** | burglar | **mesmerised** |
| leg | **purpose** | insurance | **miniature** |
| lets | question | carriage | **minions** |
| lick | quickly | interesting | **mischievous** |
| **like** | rain | irrational | **misconstrue** |
| long | region | journey | **misogyny** |

| look | report | category | recognisable |
|---|---|---|---|
| lot | **results** | **kiosk** | **narcissist** |
| may | riot | celebration | **necessary** |
| me | rodent | **literacy** | **nonchalant** |
| meet | rumble | **lyrebird** | **noticeable** |
| men | saving | **malt** | **notoriety** |
| milk | scare | **massive** | **nuisance** |
| much | school | **mayor** | obedience |
| my | scream | medieval | **obnoxious** |
| **name** | shaking | community | **obscure** |
| new | shape | complete | **observation** |
| no | shout | muscular | **obsessive** |
| not | sitting | concerned | **occasions** |
| of | sky | confidence | occasionally |
| old | sound | neither | **oscillate** |
| one | steal | **notice** | **peculiar** |
| our | stopped | **obviously** | personally |
| park | strip | circuit | **persuasive** |
| pay | suddenly | **opportunity** | **phosphorescent** |
| pen | table | optimist | **plateau** |
| play | teacher | **origami** | **population** |
| plot | their | parallel | **precise** |
| pull | sticking | pedestal | **prevalence** |
| put | train | prankster | **privileged** |
| ran | travel | precious | **proposition** |
| red | graves | presence | **psychic** |
| **rest** | **uncle** | principle | **psychology** |
| room | until | punctual | **quiescent** |
| rot | useful | pursuit | **racquet** |
| run | very | **quench** | **rancour** |
| sad | walking | coordinator | realistically |
| saw | **wall** | **realistic** | **redemption** |
| say | **warn** | corpses | pessimistic |
| see | webbed | recommend | reminiscent |
| seed | when | reluctant | responsibility |
| seem | which | **remorse** | **resurrect** |
| set | **who** | **responsible** | **ricochet** |
| shed | **wings** | creature | **rigorous** |
| shop | pushing | scavenger | **sabotage** |
| sing | windy | sceptical | **scimitar** |
| sit | **yellow** | **scientific** | **separate** |
| six | your | **shoulder** | **silhouette** |
| slow | zapped | **signal** | **sovereign** |
| so | wrong | society | stationary |
| spot | track | stammered | stationery |
| **stand** | **find** | success | telekinesis |
| sleep | boat | **suitable** | temperamental |
| teeth | named | criminal | temporary |
| tell | smashed | decorate | therapeutic |
| tells | later | temperature | **thoroughly** |

| | | | |
|---|---|---|---|
| ten | wasn't | terrace | **tournament** |
| that | won't | terrified | **tsunami** |
| the | **work** | **thermonuclear** | ubiquitous |
| then | named | treasure | **unconscious** |
| **thing** | knocked | **unexpectedly** | unnecessary |
| this | floating | **unnatural** | **vertebrates** |
| to | fainted | **useless** | **voila** |
| **today** | oval | **valuable** | resuscitate |
| top | Wednesday | vessel | **wilful** |
| **undo** | planned | curious | **wondrous** |
| vat | taking | vortex | **zephyr** |
| vet | massive | **wealthy** | **petrified** |
| was | haunted | **weighed** | frantically |
| we | **already** | whisper | desperately |
| well | anything | women | **behemoth** |
| went | truth | contraptions | **devastating** |
| will | looking | **yacht** | **assailant** |
| wish | leader | **youthful** | **reminiscing** |
| with | stole | zenith | **wielded** |
| yell | sister | **guards** | **miniscule** |
| yes | crying | fractured | **absenteeism** |
| zoo | gone | **surfaced** | absorbency |
| king | death | telescopes | abysmally |
| on | time | depression | academically |
| fox | carried | excursion | accumulative |
| pit | careful | **imagined** | **acrimonious** |
| pot | didn't | exotic | affirmatively |
| **test** | friends | scientist | embarrassingly |
| flow | **say** | **pollution** | penetrative |
| frog | didn't | horizon | serendipitous |
| his | recount | centuries | vituperative |
| pet | drifted | demolished | **virologist** |
| dish | **back** | demonstrate | vigilantism |

# Appendix C

## SpellChecker Program Results

| Essay ID | Spelling Errors | Grammar Errors | Words | Sentences | Flesch reading ease | Flesch-Kgrade level | NormSpel | NormPunc |
|---|---|---|---|---|---|---|---|---|
| Adano_35 | 26 | 9 | 452 | 29 | 80.3 | 5.9 | 0.057522 | 0.3103448 |
| Amess_33 | 7 | 14 | 465 | 37 | 86.1 | 4.3 | 0.015054 | 0.3783784 |
| Andrews_34 | 6 | 9 | 662 | 57 | 87.1 | 3.9 | 0.009063 | 0.1578947 |
| Ansell_24 | 14 | 4 | 238 | 15 | 83.3 | 5.5 | 0.058824 | 0.2666667 |
| Araya_23 | 23 | 3 | 264 | 28 | 89 | 3 | 0.087121 | 0.1071429 |
| Arney_20 | 14 | 7 | 208 | 16 | 82.2 | 4.7 | 0.067308 | 0.4375 |
| Atkinson_23 | 22 | 16 | 323 | 33 | 96.3 | 2.2 | 0.068111 | 0.4848485 |
| Azmi_36 | 0 | 1 | 132 | 10 | 87.6 | 4.3 | 0 | 0.1 |
| Bagiatis_42 | 7 | 6 | 467 | 31 | 77.6 | 6.1 | 0.014989 | 0.1935484 |
| Baker.C_45 | 1 | 0 | 362 | 20 | 79 | 6.1 | 0.002762 | 0 |
| Baker.L_45 | 22 | 6 | 608 | 48 | 86.6 | 4.3 | 0.036184 | 0.125 |
| Bate-Rowles_21 | 11 | 3 | 140 | 12 | 87.4 | 3.9 | 0.078571 | 0.25 |
| Bellis_35 | 9 | 4 | 470 | 21 | 74.8 | 8 | 0.019149 | 0.1904762 |
| Bennett_30 | 23 | 2 | 454 | 13 | 65.7 | 12.1 | 0.050661 | 0.1538462 |
| Berente_31 | 38 | 8 | 489 | 36 | 83.2 | 4.8 | 0.07771 | 0.2222222 |
| Bertola_32 | 0 | 4 | 403 | 19 | 76.1 | 7.9 | 0 | 0.2105263 |
| Betti_35 | 19 | 16 | 384 | 40 | 92.2 | 2.7 | 0.049479 | 0.4 |
| Birss_42 | 9 | 12 | 567 | 86 | 94 | 1.7 | 0.015873 | 0.1395349 |
| Boccamazzo_32 | 1 | 8 | 535 | 67 | 87.1 | 3 | 0.001869 | 0.119403 |
| Boccamazzo_39 | 12 | 9 | 379 | 41 | 91.4 | 2.7 | 0.031662 | 0.2195122 |
| Boddington_26 | 17 | 8 | 371 | 27 | 92.1 | 3.5 | 0.045822 | 0.2962963 |
| Boles-Ryan_45 | 13 | 9 | 528 | 51 | 86.9 | 3.6 | 0.024621 | 0.1764706 |
| Borg_27 | 37 | 5 | 385 | 27 | 83.5 | 5.1 | 0.096104 | 0.1851852 |
| Bothma_43 | 2 | 10 | 445 | 38 | 92 | 3.3 | 0.004494 | 0.2631579 |
| Both-Watson_45 | 26 | 6 | 372 | 33 | 67.7 | 6.4 | 0.069892 | 0.1818182 |
| Bowen_36 | 14 | 5 | 463 | 44 | 83.9 | 4.1 | 0.030238 | 0.1136364 |
| Boyle_29 | 6 | 3 | 341 | 16 | 73.5 | 8.2 | 0.017595 | 0.1875 |
| Brampton_33 | 8 | 8 | 456 | 26 | 80.4 | 5.6 | 0.017544 | 0.3076923 |
| Brean_43 | 3 | 8 | 490 | 51 | 70.6 | 5.7 | 0.006122 | 0.1568627 |
| Byrnes_34 | 10 | 12 | 421 | 37 | 87.6 | 3.8 | 0.023753 | 0.3243243 |
| Cabunalda_41 | 11 | 7 | 458 | 37 | 85.1 | 4.4 | 0.024017 | 0.1891892 |
| Campbell_24 | 22 | 12 | 389 | 24 | 91.9 | 4.2 | 0.056555 | 0.5 |
| Castaing_40 | 1 | 4 | 522 | 49 | 89.3 | 3.4 | 0.001916 | 0.0816327 |
| Catovic_38 | 5 | 8 | 389 | 29 | 73.1 | 6.2 | 0.012853 | 0.2758621 |
| Chandler_37 | 8 | 5 | 435 | 28 | 85.1 | 5.1 | 0.018391 | 0.1785714 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Charles_31 | 3 | 1 | 360 | 51 | 83.3 | 3.3 | 0.008333 | 0.0196078 |
| Chedid_44 | 4 | 10 | 646 | 63 | 90.5 | 3 | 0.006192 | 0.1587302 |
| Chu_40 | 2 | 9 | 556 | 53 | 89.5 | 3.3 | 0.003597 | 0.1698113 |
| Cockman_26 | 8 | 4 | 258 | 14 | 86 | 5.2 | 0.031008 | 0.2857143 |
| Colby_31 | 10 | 5 | 308 | 22 | 94 | 3.6 | 0.032468 | 0.2272727 |
| Cole_25 | 12 | 8 | 378 | 34 | 79.7 | 4.8 | 0.031746 | 0.2352941 |
| Conn_33 | 17 | 7 | 359 | 26 | 77.1 | 5.9 | 0.047354 | 0.2692308 |
| Coppard_37 | 15 | 11 | 419 | 33 | 85.7 | 4.4 | 0.0358 | 0.3333333 |
| Corfias_29 | 21 | 22 | 475 | 36 | 87.9 | 4.1 | 0.044211 | 0.6111111 |
| Cowell_33 | 10 | 14 | 576 | 35 | 86.3 | 5 | 0.017361 | 0.4 |
| Cunning-ham_39 | 2 | 4 | 400 | 32 | 87.2 | 4 | 0.005 | 0.125 |
| Dalton_28 | 10 | 11 | 410 | 19 | 83.8 | 6.8 | 0.02439 | 0.5789474 |
| Danks_42 | 5 | 11 | 400 | 33 | 90 | 3.7 | 0.0125 | 0.3333333 |
| Day-Dressa_26 | 5 | 4 | 312 | 8 | 59.4 | 14.5 | 0.016026 | 0.5 |
| De Melo_36 | 2 | 8 | 482 | 48 | 87.3 | 3.5 | 0.004149 | 0.1666667 |
| De Pledge_38 | 13 | 11 | 464 | 46 | 92.4 | 2.8 | 0.028017 | 0.2391304 |
| Dickerson_24 | 30 | 8 | 366 | 24 | 85.9 | 5 | 0.081967 | 0.3333333 |
| Dixon_39 | 8 | 16 | 488 | 51 | 82.7 | 4.1 | 0.016393 | 0.3137255 |
| Doran_43 | 11 | 6 | 435 | 33 | 76.9 | 5.7 | 0.025287 | 0.1818182 |
| Dorrell_33 | 8 | 9 | 472 | 41 | 91.3 | 3.3 | 0.016949 | 0.2195122 |
| Dreja_26 | 21 | 13 | 251 | 21 | 86.3 | 4.1 | 0.083665 | 0.6190476 |
| Effendi_25 | 10 | 15 | 256 | 27 | 86.8 | 3.5 | 0.039063 | 0.5555556 |
| Eisenlohr_37 | 21 | 16 | 420 | 53 | 85 | 3.1 | 0.05 | 0.3018868 |
| Estens_30 | 21 | 9 | 448 | 27 | 83 | 5.6 | 0.046875 | 0.3333333 |
| Felix_26 | 8 | 8 | 218 | 28 | 90.2 | 2.6 | 0.036697 | 0.2857143 |
| Foo_46 | 1 | 5 | 728 | 94 | 89 | 2.6 | 0.001374 | 0.0531915 |
| Forward_31 | 27 | 10 | 485 | 37 | 80.9 | 5.1 | 0.05567 | 0.2702703 |
| French_25 | 6 | 6 | 171 | 11 | 76.2 | 6.4 | 0.035088 | 0.5454545 |
| Galante_30 | 7 | 16 | 494 | 48 | 87.4 | 3.6 | 0.01417 | 0.3333333 |
| Gaunt_43 | 8 | 4 | 538 | 42 | 82.7 | 4.8 | 0.01487 | 0.0952381 |
| Gazeley_25 | 4 | 4 | 161 | 8 | 84.9 | 6.4 | 0.024845 | 0.5 |
| Geary_24 | 14 | 13 | 424 | 30 | 88.5 | 4.4 | 0.033019 | 0.4333333 |
| Geste_42 | 8 | 10 | 480 | 52 | 92.5 | 2.5 | 0.016667 | 0.1923077 |
| Gianatti_31 | 17 | 13 | 572 | 43 | 81.9 | 5 | 0.02972 | 0.3023256 |
| Gorjy_38 | 14 | 13 | 375 | 22 | 87 | 4.8 | 0.037333 | 0.5909091 |
| Graeser_34 | 7 | 13 | 432 | 37 | 87.9 | 3.7 | 0.016204 | 0.3513514 |
| Graham_31 | 4 | 9 | 314 | 20 | 86.5 | 5 | 0.012739 | 0.45 |
| Griffiths_27 | 4 | 1 | 161 | 6 | 57.5 | 11.8 | 0.024845 | 0.1666667 |
| Grynychyn_36 | 5 | 6 | 623 | 65 | 88.8 | 3.1 | 0.008026 | 0.0923077 |
| Haines_39 | 21 | 14 | 473 | 55 | 88.6 | 3 | 0.044397 | 0.2545455 |
| Harrison_27 | 13 | 13 | 314 | 29 | 74.8 | 5.4 | 0.041401 | 0.4482759 |
| Hart_30 | 24 | 11 | 502 | 33 | 87.2 | 4.8 | 0.047809 | 0.3333333 |
| Hawkett_45 | 9 | 12 | 535 | 41 | 86 | 4.4 | 0.016822 | 0.2926829 |
| Haynes_32 | 13 | 9 | 331 | 40 | 89.1 | 2.8 | 0.039275 | 0.225 |
| Helsby_31 | 18 | 18 | 502 | 45 | 88.6 | 3.3 | 0.035857 | 0.4 |
| Heremia_41 | 6 | 7 | 437 | 33 | 83.6 | 4.5 | 0.01373 | 0.2121212 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Hermansyah_34 | 17 | 13 | 509 | 52 | 85 | 3.5 | 0.033399 | 0.25 |
| Herriman_46 | 7 | 8 | 400 | 24 | 78.6 | 5.9 | 0.0175 | 0.3333333 |
| Higginson_30 | 7 | 9 | 403 | 32 | 86.5 | 4.2 | 0.01737 | 0.28125 |
| Hocking_34 | 8 | 3 | 387 | 21 | 73.1 | 7.6 | 0.020672 | 0.1428571 |
| Holst_28 | 15 | 8 | 305 | 12 | 74.2 | 9.1 | 0.04918 | 0.6666667 |
| Holt_33 | 5 | 5 | 509 | 37 | 83.6 | 5 | 0.009823 | 0.1351351 |
| Hughes_30 | 19 | 6 | 416 | 37 | 81 | 4.7 | 0.045673 | 0.1621622 |
| Hulbert_43 | 6 | 10 | 278 | 37 | 92.9 | 2.1 | 0.021583 | 0.2702703 |
| Ingle_46 | 16 | 5 | 513 | 28 | 75.2 | 7.3 | 0.031189 | 0.1785714 |
| Ingram_30 | 49 | 17 | 531 | 23 | 86.7 | 6.8 | 0.092279 | 0.7391304 |
| Jambor_32 | 5 | 6 | 429 | 31 | 92.8 | 3.4 | 0.011655 | 0.1935484 |
| Johnson_36 | 9 | 10 | 503 | 20 | 75.1 | 9 | 0.017893 | 0.5 |
| Jones_33 | 9 | 12 | 383 | 34 | 90.1 | 3.1 | 0.023499 | 0.3529412 |
| Jurgenson_47 | 6 | 4 | 532 | 51 | 73.1 | 5.5 | 0.011278 | 0.0784314 |
| Karski_37 | 2 | 4 | 298 | 35 | 83.2 | 3.7 | 0.006711 | 0.1142857 |
| Kelly_26 | 11 | 6 | 211 | 12 | 89.6 | 4.7 | 0.052133 | 0.5 |
| Kerr_38 | 9 | 10 | 369 | 21 | 77.4 | 6.7 | 0.02439 | 0.4761905 |
| Kershaw_25 | 14 | 5 | 201 | 11 | 85.5 | 5.8 | 0.069652 | 0.4545455 |
| Kroll_43 | 0 | 6 | 421 | 39 | 90.4 | 3.2 | 0 | 0.1538462 |
| Lee_26 | 11 | 1 | 206 | 8 | 77.3 | 8.7 | 0.053398 | 0.125 |
| Leslie_42 | 6 | 5 | 661 | 34 | 74.4 | 7.4 | 0.009077 | 0.1470588 |
| Lewis_34 | 20 | 14 | 440 | 48 | 94 | 2.4 | 0.045455 | 0.2916667 |
| Lim_28 | 11 | 12 | 404 | 44 | 96.5 | 2 | 0.027228 | 0.2727273 |
| Lloyd_39 | 14 | 11 | 438 | 19 | 78.4 | 7.5 | 0.031963 | 0.5789474 |
| Logan_40 | 13 | 14 | 486 | 42 | 86.3 | 3.9 | 0.026749 | 0.3333333 |
| Loh_46 | 4 | 9 | 561 | 40 | 76.7 | 6 | 0.00713 | 0.225 |
| Lungu_29 | 15 | 9 | 613 | 13 | 57.9 | 16.6 | 0.02447 | 0.6923077 |
| MacKenzie_28 | 55 | 27 | 645 | 58 | 92.9 | 2.9 | 0.085271 | 0.4655172 |
| Madadi_35 | 17 | 13 | 490 | 34 | 86.6 | 4.5 | 0.034694 | 0.3823529 |
| Main_44 | 3 | 12 | 616 | 31 | 74.1 | 6.8 | 0.00487 | 0.3870968 |
| Marchant_37 | 6 | 6 | 295 | 38 | 89.6 | 2.4 | 0.020339 | 0.1578947 |
| Marsh_27 | 11 | 7 | 449 | 20 | 72.6 | 8.6 | 0.024499 | 0.35 |
| Mason_29 | 11 | 7 | 236 | 21 | 85.4 | 4.1 | 0.04661 | 0.3333333 |
| Massey_41 | 7 | 9 | 515 | 32 | 83.5 | 5.6 | 0.013592 | 0.28125 |
| Mawer_40 | 6 | 8 | 597 | 62 | 87.8 | 3.3 | 0.01005 | 0.1290323 |
| McBeath_41 | 15 | 8 | 314 | 24 | 87.1 | 4.1 | 0.047771 | 0.3333333 |
| McCleery_38 | 2 | 16 | 412 | 44 | 91.3 | 2.7 | 0.004854 | 0.3636364 |
| McDermott_23 | 15 | 5 | 426 | 23 | 90.9 | 5.1 | 0.035211 | 0.2173913 |
| McFarlane_32 | 31 | 5 | 341 | 9 | 76.4 | 10.7 | 0.090909 | 0.5555556 |
| McGuire_39 | 13 | 6 | 421 | 42 | 68.6 | 6.1 | 0.030879 | 0.1428571 |
| McInnes_40 | 2 | 9 | 374 | 33 | 84 | 4.3 | 0.005348 | 0.2727273 |
| Menezes_46 | 3 | 10 | 458 | 45 | 93.6 | 2.7 | 0.00655 | 0.2222222 |
| Merritt_35 | 0 | 7 | 413 | 19 | 82.4 | 6.8 | 0 | 0.3684211 |
| Metcalfe_46 | 0 | 5 | 596 | 59 | 80 | 4.6 | 0 | 0.0847458 |
| Mitrevski_28 | 19 | 5 | 246 | 21 | 83.2 | 4.5 | 0.077236 | 0.2380952 |
| Mokrzycki_32 | 15 | 4 | 320 | 19 | 81 | 5.9 | 0.046875 | 0.2105263 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Naderi_32 | 13 | 5 | 358 | 21 | 79.1 | 6.4 | 0.036313 | 0.2380952 |
| Naschwitz_38 | 4 | 6 | 307 | 23 | 80.3 | 5.3 | 0.013029 | 0.2608696 |
| Ngo_20 | 13 | 5 | 271 | 12 | 80.2 | 7.6 | 0.04797 | 0.4166667 |
| Nguyen_46 | 2 | 7 | 711 | 62 | 88.6 | 3.6 | 0.002813 | 0.1129032 |
| Palayukan_43 | 12 | 3 | 410 | 37 | 78.5 | 5 | 0.029268 | 0.0810811 |
| Persson_44 | 4 | 4 | 553 | 41 | 81.7 | 5.2 | 0.007233 | 0.097561 |
| Pickering_44 | 31 | 9 | 429 | 30 | 71.8 | 6.6 | 0.072261 | 0.3 |
| Pickett_29 | 5 | 3 | 238 | 8 | 67.8 | 11.1 | 0.021008 | 0.375 |
| Pidgeon_34 | 17 | 16 | 679 | 26 | 70.9 | 9.5 | 0.025037 | 0.6153846 |
| Pizzirani_27 | 11 | 11 | 339 | 20 | 84.5 | 5.6 | 0.032448 | 0.55 |
| Pugh_42 | 10 | 10 | 609 | 33 | 77.9 | 6.9 | 0.01642 | 0.3030303 |
| Ransom_29 | 16 | 7 | 302 | 23 | 79.7 | 5.3 | 0.05298 | 0.3043478 |
| Reid_23 | 1 | 4 | 263 | 12 | 80.6 | 7.4 | 0.003802 | 0.3333333 |
| Ryan_27 | 2 | 4 | 357 | 21 | 80.8 | 6.1 | 0.005602 | 0.1904762 |
| Saurin_36 | 6 | 9 | 437 | 50 | 95.4 | 2 | 0.01373 | 0.18 |
| Sekhon_47 | 12 | 6 | 545 | 47 | 83.5 | 4.4 | 0.022018 | 0.1276596 |
| Shapland_44 | 2 | 5 | 574 | 50 | 92.4 | 3.1 | 0.003484 | 0.1 |
| Sharma_39 | 0 | 6 | 527 | 44 | 82.3 | 4.6 | 0 | 0.1363636 |
| Sharpe_47 | 6 | 7 | 575 | 38 | 72 | 6.8 | 0.010435 | 0.1842105 |
| Shelton_35 | 10 | 15 | 443 | 32 | 88.1 | 4.2 | 0.022573 | 0.46875 |
| Shook_41 | 7 | 5 | 366 | 26 | 74.4 | 6.3 | 0.019126 | 0.1923077 |
| Slaughter_36 | 21 | 3 | 340 | 16 | 74 | 8 | 0.061765 | 0.1875 |
| Somas_35 | 21 | 14 | 427 | 55 | 95.5 | 1.8 | 0.04918 | 0.2545455 |
| Stevens_47 | 10 | 9 | 421 | 48 | 100 | 0.6 | 0.023753 | 0.1875 |
| Stewart_46 | 12 | 5 | 462 | 47 | 77.3 | 4.8 | 0.025974 | 0.106383 |
| Sutikno_38 | 14 | 3 | 300 | 19 | 74.4 | 6.7 | 0.046667 | 0.1578947 |
| Sutton_37 | 13 | 9 | 525 | 32 | 71.2 | 7.3 | 0.024762 | 0.28125 |
| Tan_47 | 2 | 3 | 404 | 39 | 82.8 | 4.2 | 0.00495 | 0.0769231 |
| Tarawa_23 | 19 | 11 | 256 | 19 | 91.7 | 3.3 | 0.074219 | 0.5789474 |
| Taylor_42 | 12 | 16 | 686 | 26 | 78.3 | 8.1 | 0.017493 | 0.6153846 |
| Taylor_45 | 2 | 6 | 424 | 27 | 71.1 | 7.1 | 0.004717 | 0.2222222 |
| Temelcos_27 | 16 | 9 | 443 | 23 | 77.4 | 7 | 0.036117 | 0.3913043 |
| Teremoana_17 | 13 | 3 | 270 | 6 | 63.5 | 14.5 | 0.048148 | 0.5 |
| Thomas_28 | 17 | 10 | 351 | 26 | 87.1 | 4.4 | 0.048433 | 0.3846154 |
| Truell_45 | 2 | 6 | 317 | 25 | 85.1 | 4.4 | 0.006309 | 0.24 |
| Van der Meer_41 | 12 | 14 | 479 | 34 | 81.3 | 5.3 | 0.025052 | 0.4117647 |
| Van Noort_41 | 3 | 7 | 463 | 35 | 86.3 | 4.5 | 0.006479 | 0.2 |
| Vickery_47 | 2 | 4 | 425 | 34 | 78.6 | 5.3 | 0.004706 | 0.1176471 |
| Wilford_19 | 17 | 4 | 235 | 12 | 93.2 | 4.9 | 0.07234 | 0.3333333 |
| Williams_25 | 20 | 10 | 316 | 29 | 89.1 | 3.5 | 0.063291 | 0.3448276 |
| Williams_29 | 23 | 9 | 435 | 22 | 85.6 | 6.2 | 0.052874 | 0.4090909 |
| Williams_40 | 4 | 7 | 457 | 31 | 89.1 | 4.4 | 0.008753 | 0.2258065 |
| Wilson_40 | 3 | 8 | 512 | 67 | 91.9 | 2.3 | 0.005859 | 0.119403 |
| Wright_44 | 11 | 7 | 584 | 32 | 84.2 | 5.9 | 0.018836 | 0.21875 |
| Wych_37 | 5 | 13 | 479 | 44 | 90.3 | 3.3 | 0.010438 | 0.2954545 |

# Appendix D

## Algorithm for grading vocabulary in poor essays – Results

| Essay ID | CW% | Total words | Target | Result |
|---|---|---|---|---|
| Agenbag_23 | 0.417323 | 208 | 2 | 2 |
| Allanson_8 | 0.211538 | 219 | 2 | 1 |
| Anderson_7 | 0.212121 | 48 | 1 | 1 |
| Batty_9 | 0.466667 | 17 | 1 | 1 |
| Beaven_20 | 0.625 | 276 | 2 | 2 |
| Bekisz_25 | 0.525974 | 263 | 2 | 2 |
| Bonney_6 | 0.222222 | 10 | 1 | 1 |
| Borger_21 | 0.45 | 150 | 2 | 2 |
| Bropho_2 | 0.5 | 4 | 1 | 1 |
| Brovadan_1 | 0.5 | 2 | 1 | 1 |
| Burazin-Pense_28 | 0.456693 | 235 | 2 | 2 |
| Burns_10 | 0.666667 | 30 | 1 | 1 |
| Callow_3 | 0.571429 | 8 | 1 | 1 |
| Campbell_21 | 0.5625 | 116 | 2 | 2 |
| Carter_2 | 0 | 1 | 0 | 0 |
| Cherel_14 | 0.710526 | 80 | 2 | 2 |
| Chestnut_9 | 0.714286 | 17 | 1 | 1 |
| Chetwynd_16 | 0.549451 | 203 | 2 | 2 |
| Combi_20 | 0.581395 | 165 | 2 | 2 |
| Cotchin_6 | 0.727273 | 12 | 1 | 1 |
| Cox_11 | 0.549451 | 220 | 2 | 2 |
| Coyne_15 | 0.673077 | 80 | 2 | 2 |
| Cubbin_2 | 0.6875 | 16 | 1 | 1 |
| Cunningham_10 | 0.391304 | 30 | 2 | 1 |
| Dale-Fraser_11 | 0.741935 | 45 | 2 | 2 |
| Dann_4 | 0.173913 | 153 | 1 | 1 |
| Darcey_19 | 0.482759 | 163 | 2 | 2 |
| Deliu_18 | 0.425532 | 140 | 2 | 2 |
| Dewar_10 | 0.589744 | 80 | 1 | 2 |
| Dodd_21 | 0.522727 | 317 | 2 | 2 |
| Dopoe_12 | 0.425926 | 94 | 2 | 2 |
| Dorant_24 | 0.412429 | 371 | 2 | 2 |
| Eckerman_3 | 0.454545 | 14 | 1 | 1 |
| Ellerton_16 | 0.302326 | 160 | 2 | 1 |

| | | | | |
|---|---|---|---|---|
| Erturk_22 | 0.544554 | 208 | 2 | 2 |
| Farley_19 | 0.450549 | 164 | 2 | 2 |
| Farrell_12 | 0.473684 | 21 | 1 | 1 |
| Ferguson_20 | 0.513158 | 107 | 2 | 2 |
| Fermaner_6 | 0.5 | 10 | 1 | 1 |
| Fischer_24 | 0.484127 | 243 | 2 | 2 |
| Gandy_9 | 0.65625 | 62 | 2 | 2 |
| Glazebrook_8 | 0.285714 | 48 | 1 | 1 |
| Goodison_4 | 0.75 | 4 | 1 | 1 |
| Green_5 | 0.833333 | 6 | 1 | 1 |
| Guthrie_14 | 0.425287 | 130 | 2 | 2 |
| Hague_13 | 0.509091 | 112 | 2 | 2 |
| Hall_22 | 0.384615 | 154 | 2 | 1 |
| Hansen_11 | 0.521739 | 24 | 2 | 1 |
| Hansen_19 | 0.642857 | 119 | 2 | 2 |
| Harland_15 | 0.540541 | 48 | 2 | 2 |
| Heal_5 | 0.666667 | 37 | 1 | 2 |
| Henry_13 | 0.674419 | 54 | 2 | 2 |
| Hill_2 | 0.166667 | 24 | 1 | 1 |
| Hodson_18 | 0.4 | 117 | 2 | 2 |
| Hudson_21 | 0.395349 | 154 | 2 | 1 |
| Hughes_1 | 0.357143 | 14 | 1 | 1 |
| Hunter_16 | 0.27451 | 71 | 2 | 1 |
| Ingram_7 | 0.857143 | 7 | 1 | 1 |
| Ioppolo_21 | 0.511628 | 201 | 2 | 2 |
| Jones_23 | 0.394737 | 272 | 2 | 1 |
| Jones_3 | 0.428571 | 7 | 1 | 1 |
| Jovanovic_16 | 0.55914 | 263 | 2 | 2 |
| Kamara_15 | 0.571429 | 267 | 2 | 2 |
| Kelly_15 | 0.461538 | 68 | 2 | 2 |
| Kirkegaard_4 | 0.2 | 20 | 1 | 1 |
| Knights_15 | 0.464567 | 191 | 2 | 2 |
| Kodi_18 | 0.522388 | 197 | 2 | 2 |
| Lloyd_18 | 0.430769 | 179 | 2 | 2 |
| MacDonald_12 | 0.411765 | 71 | 2 | 2 |
| Mason_14 | 0.509804 | 219 | 2 | 2 |
| Maya_17 | 0.534653 | 159 | 2 | 2 |
| Mclean_3 | 0.5 | 4 | 1 | 1 |
| Merritt_19 | 0.45 | 203 | 2 | 2 |
| Milton_22 | 0.453846 | 219 | 2 | 2 |
| Mongoo_8 | 0.5 | 15 | 1 | 1 |
| Naismith_7 | 0.127451 | 124 | 1 | 1 |
| Ninyette_13 | 0.6 | 91 | 1 | 2 |
| Nshuti_21 | 0.47619 | 224 | 2 | 2 |
| O'Brien_14 | 0.39604 | 174 | 2 | 1 |
| Olliver_10 | 0.545455 | 61 | 1 | 2 |

| | | | | |
|---|---|---|---|---|
| Peck_1 | 0.5 | 2 | 0 | 1 |
| Phophimol_14 | 0.727273 | 182 | 2 | 2 |
| Piper_7 | 0.307692 | 15 | 1 | 1 |
| Pizzey_12 | 0.603604 | 299 | 2 | 2 |
| Polak_7 | 0.473684 | 39 | 1 | 2 |
| Porter_5 | 0.184211 | 55 | 2 | 1 |
| Potter_9 | 0.206349 | 90 | 1 | 1 |
| Pritchard_8 | 0.405405 | 53 | 1 | 2 |
| Pryor_12 | 0.512195 | 86 | 2 | 2 |
| Purves_14 | 0.492063 | 117 | 2 | 2 |
| Quartermaine_10 | 0.5 | 35 | 1 | 2 |
| Rieger_4 | 0.5 | 6 | 1 | 1 |
| Roberts_8 | 0.2 | 5 | 1 | 1 |
| Rosewood_16 | 0.654545 | 138 | 2 | 2 |
| Rumball_6 | 0.178571 | 76 | 1 | 1 |
| Ryan_8 | 0.714286 | 9 | 1 | 1 |
| Schnaars_6 | 0.666667 | 6 | 1 | 1 |
| Sears_1 | 0 | 1 | 0 | 0 |
| Shannon_10 | 0.578947 | 20 | 1 | 1 |
| Shea_3 | 0.076923 | 13 | 0 | 1 |
| Simpson_3 | 0.5 | 4 | 1 | 1 |
| Simpson_6 | 0.25 | 9 | 1 | 1 |
| Springer_15 | 0.552239 | 304 | 2 | 2 |
| Strudwick_7 | 0.5 | 4 | 1 | 1 |
| Swan_19 | 0.528302 | 223 | 2 | 2 |
| Taniwha_4 | 0.357143 | 131 | 1 | 1 |
| Taylor_18 | 0.543478 | 89 | 2 | 2 |
| Tognolini_5 | 0.1 | 40 | 1 | 1 |
| Tuck_19 | 0.536232 | 114 | 2 | 2 |
| Ugle_11 | 0.588235 | 52 | 2 | 2 |
| Ugle_6 | 0.75 | 4 | 1 | 1 |
| Varischetti_11 | 0.421053 | 23 | 1 | 1 |
| Vaughan_17 | 0.517241 | 136 | 2 | 2 |
| Verwey_12 | 0.421053 | 47 | 2 | 2 |
| Vida_17 | 0.541985 | 322 | 2 | 2 |
| Vincent_10 | 0.75 | 49 | 1 | 2 |
| Walley_8 | 0.166667 | 7 | 1 | 1 |
| Walley_9 | 0.47619 | 30 | 1 | 1 |
| Warner_18 | 0.494624 | 202 | 2 | 2 |
| Whitby_4 | 0.6 | 5 | 0 | 1 |
| White_5 | 0.333333 | 3 | 1 | 1 |
| Wicks_22 | 0.548673 | 201 | 2 | 2 |
| Williams_18 | 0.404762 | 153 | 2 | 2 |
| Willis_13 | 0.518519 | 34 | 2 | 2 |
| Wood_17 | 0.346154 | 129 | 2 | 1 |
| Woodhouse_2 | 0.469697 | 34 | 1 | 2 |

| | | | | |
|---|---|---|---|---|
| Wright_13 | 0.407407 | 70 | 2 | 2 |
| Wyld_20 | 0.461538 | 111 | 2 | 2 |
| Ybanez_12 | 0.2 | 5 | 1 | 1 |
| Youens_9 | 0.285714 | 7 | 1 | 1 |
| Zico_15 | 0.69863 | 155 | 2 | 2 |

# Appendix E

## Algorithm for grading sentence structure in poor essays – Results

| EssayID | Target | Result |
|---|---|---|
| Adkins_24.doc | 2 | 1 |
| Agenbag_23.doc | 2 | 3 |
| Batty_9.doc | 1 | 1 |
| Beaven_20.doc | 2 | 3 |
| Bekisz_25.doc | 3 | 2 |
| Borger_21.doc | 2 | 3 |
| Bropho_2.doc | 0 | 0 |
| Brovadan_1.doc | 0 | 0 |
| Burazin-Pense_28.doc | 2 | 2 |
| Burns_10.doc | 1 | 1 |
| Callow_3.doc | 0 | 1 |
| Campbell_21.doc | 3 | 3 |
| Cherel_14.doc | 0 | 2 |
| Chestnut_9.doc | 1 | 2 |
| Chetwynd_16.doc | 2 | 2 |
| Combi_20.doc | 2 | 2 |
| Cotchin_6.doc | 1 | 1 |
| Cox_11.doc | 1 | 3 |
| Coyne_15.doc | 2 | 1 |
| Cubbin_2.doc | 0 | 1 |
| Cunningham_10.doc | 1 | 1 |
| Dale-Fraser_11.doc | 2 | 1 |
| Darcey_19.doc | 3 | 3 |
| Deliu_18.doc | 3 | 2 |
| Dewar_10.doc | 1 | 2 |
| Dodd_21.doc | 2 | 1 |
| Dopoe_12.doc | 2 | 1 |
| Dorant_24.doc | 2 | 3 |
| Eckerman_3.doc | 1 | 2 |
| Ellerton_16.doc | 1 | 2 |
| Erturk_22.doc | 3 | 2 |
| Farley_19.doc | 2 | 2 |
| Farrell_12.doc | 1 | 1 |

| | | |
|---|---|---|
| Ferguson_20.doc | 3 | 2 |
| Fermaner_6.doc | 1 | 1 |
| Fischer_24.doc | 3 | 2 |
| Gandy_9.doc | 2 | 2 |
| Glazebrook_8.doc | 1 | 2 |
| Goodison_4.doc | 1 | 0 |
| Green_5.doc | 0 | 0 |
| Guthrie_14.doc | 2 | 1 |
| Hague_13.doc | 1 | 2 |
| Hall_22.doc | 2 | 2 |
| Hansen_11.doc | 2 | 1 |
| Hansen_19.doc | 2 | 2 |
| Harland_15.doc | 3 | 2 |
| Heal_5.doc | 1 | 1 |
| Henry_13.doc | 2 | 2 |
| Hodson_18.doc | 2 | 2 |
| Homewood_13.doc | 2 | 2 |
| Hudson_21.doc | 2 | 2 |
| Hughes_1.doc | 0 | 2 |
| Hunter_16.doc | 2 | 2 |
| Ingram_7.doc | 0 | 1 |
| Ioppolo_21.doc | 2 | 2 |
| Jones_23.doc | 2 | 3 |
| Jones_3.doc | 0 | 0 |
| Jovanovic_16.doc | 2 | 2 |
| Kamara_15.doc | 2 | 2 |
| Kelly_15.doc | 2 | 1 |
| Knights_15.doc | 2 | 3 |
| Kodi_18.doc | 2 | 2 |
| Lantang_20.doc | 2 | 3 |
| Lloyd_18.doc | 3 | 1 |
| MacDonald_12.doc | 2 | 2 |
| Mason_14.doc | 2 | 2 |
| Maya_17.doc | 3 | 3 |
| Mclean_3.doc | 0 | 0 |
| Merritt_19.doc | 2 | 3 |
| Milton_22.doc | 2 | 3 |
| Mongoo_8.doc | 1 | 1 |
| Ninyette_13.doc | 2 | 2 |
| Nshuti_21.doc | 2 | 3 |
| O'Brien_14.doc | 3 | 2 |
| Olliver_10.doc | 1 | 3 |
| Peck_1.doc | 0 | 0 |
| Phophimol_14.doc | 2 | 2 |
| Piper_7.doc | 1 | 1 |

| | | |
|---|---|---|
| Pizzey_12.doc | 2 | 2 |
| Polak_7.doc | 1 | 2 |
| Pritchard_8.doc | 1 | 2 |
| Pryor_12.doc | 1 | 2 |
| Purves_14.doc | 2 | 2 |
| Quartermaine_10.doc | 0 | 2 |
| Rieger_4.doc | 1 | 1 |
| Rosewood_16.doc | 2 | 2 |
| Ryan_8.doc | 1 | 0 |
| Schnaars_6.doc | 1 | 1 |
| Shannon_10.doc | 1 | 2 |
| Simpson_3.doc | 0 | 1 |
| Simpson_6.doc | 1 | 1 |
| Springer_15.doc | 2 | 2 |
| Strudwick_7.doc | 0 | 1 |
| Swan_19.doc | 2 | 2 |
| Taniwha_4.doc | 1 | 2 |
| Taylor_18.doc | 2 | 2 |
| Tuck_19.doc | 2 | 2 |
| Ugle_11.doc | 1 | 1 |
| Ugle_6.doc | 0 | 0 |
| Varischetti_11.doc | 2 | 2 |
| Vaughan_17.doc | 2 | 2 |
| Verwey_12.doc | 1 | 1 |
| Vida_17.doc | 3 | 2 |
| Vincent_10.doc | 1 | 1 |
| Walley_9.doc | 2 | 1 |
| Warner_18.doc | 3 | 2 |
| Whitby_4.doc | 0 | 1 |
| White_5.doc | 0 | 0 |
| Wicks_22.doc | 2 | 2 |
| Williams_18.doc | 2 | 3 |
| Willis_13.doc | 1 | 1 |
| Woodhouse_2.doc | 2 | 0 |
| Wood_17.doc | 0 | 2 |
| Wright_13.doc | 3 | 3 |
| Wyld_20.doc | 2 | 1 |
| Youens_9.doc | 1 | 1 |
| Zico_15.doc | 2 | 1 |