Semi-Automatic Essay Assessment based on a flexible Rubric

Master's Thesis
at
Graz University of Technology

submitted by
Andreas Weinberger
May 12, 2011

Supervisor
Uni. Doz. Dipl.-Ing. Dr. techn. Christian Gütl
IICM, Graz University of Technology

Co-Supervisor
Prof. Dr. Heinz Dreher
IS, Curtin University

Institute for Information Systems and Computer Media (IICM)
Graz University of Technology
A-8010 Graz, Austria

Computer-unterstützte Aufsatz Beurteilung basierend auf flexiblen Rubriken

Masterarbeit
an der
TU Graz

eingereicht von
Andreas Weinberger
12. Mai 2011

Betreuer
Uni. Doz. Dipl.-Ing. Dr. techn. Christian Gütl
IICM, TU Graz

Mit-Betreuer
Prof. Dr. Heinz Dreher
IS, Curtin University

Institut für Informationssyteme und Computer Medien (IICM)
TU Graz
8010 Graz, Österreich

Abstract

Assessment is an essential part of the learning process, especially in formative learning settings. Traditional assessment often serves administrative needs in the form of summative assessment. Both types of assessment are challenging as it can be difficult to ensure consistency, reliability and absence of bias. In formative assessment the problem of workload and timely results is even greater, as the task is carried out more frequently. Information technology is able to assist teachers in these challenges.

Depending on the assessment goal different test items are available. The essay test item is known to be used to train language skills and to acquire foreign languages, however, its application is broader. This is due to the fact that it is a suitable test item to test for higher order skills and therefore can be applied in a great variety of subjects and can also be used at the higher education level. Evaluating essays is a time-consuming task and therefore supporting technologies can deliver great advantages.

This Master's thesis first discusses the relevant theoretical background in detail. The second part of the thesis documents the development of a prototype system to assist teachers in essay grading and evaluation of the prototype. The prototype is designed as a semi-automatic system aiding teachers in assessment and is therefore applicable at the classroom level with low student numbers where few systems exist. The semi-automatic system is based on a flexible analytical rubric, which allows to be used the system in different educational settings and on any educational level. Teachers are supported through essay analysis and possible automatic rating suggestions.

Kurzfassung

Beurteilung ist ein integraler Bestandteil des Lernprozesses in formativen Lernumgebungen. Traditionelle Beurteilungen basieren oft auf administrativen Anforderungen besonders in summativer Form. Bei beiden Typen der Beurteilung ist es anspruchsvoll Beständigkeit, Verlässlichkeit und Ausschluss von Voreingenommenheit sicherzustellen. Bei formativer Beurteilung sind aufgrund der häufigeren Durchführung die Probleme durch erhöhten Arbeitsaufwand und erwarteten zeitgerechten Ergebnissen noch größer. Informationstechnologien können Lehrpersonal bei der Bewältigung dieser Aufgabe sinnvoll unterstützen.

Abhängig vom Ziel der Beurteilung gibt es verschiedene Aufgabenstellungen zur Auswahl. Aufsätze sind eine bekannte Möglichkeit um Sprachfertigkeiten zu trainieren und Fremdsprachen zu erwerben. Die Anwendendbarkeit von Aufsätzen ist aber viel breiter. Dies basiert auf der Tatsache, dass diese Aufgabenstellung dazu geeignet ist auch Fähigkeiten komplexerer Ordnung zu testen und daher in einer Vielzahl an Fachgebieten und auf jedem Niveau bis zur Universität verwendet werden kann. Die Beurteilung von Aufsätzen ist zeitintensiv wodurch Assistenztechnologien große Vorteile bieten können.

Diese Masterarbeit beginnt mit einer ausführlichen Diskussion der relevanten theoretischen Hintergründe. Der zweite Teil dokumentiert die Entwicklung einen Prototypsystems zur Unterstützung von Lehrenden. bei der Beurteilung von Aufsätzen. Der Prototyp wurde als halbautomatisches System zur Unterstützung entworfen und kann daher im kleineren Klassenverband eingesetzt werden wo es bisher nur wenige Systeme gibt. Das System verwendet als zentrales Element eine flexible Rubrik wodurch es in verschiedenen Lernumgebungen und auf jedem Ausbildungsniveau einsetzbar ist. Lehrende werden durch die Analyse von Aufsätzen und möglichen automatischen Bewertungsvorschlägen bei der Beurteilung unterstützt.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....
date (signature)

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

Acknowledgements

I would first like to thank my supervisor Christian Gütl for his continuous support and advice during the formation of my thesis. Due to his connections to Curtin University, I had the chance to start my thesis abroad in Perth. I therefore want to thank the entire staff at Curtin Business School and notably Dr. Heinz Dreher, my co-supervisor during my time in Perth. I also want to thank Dr. Vanessa Chang, head of School of Information Systems at Curtin University, who enabled my stay at Curtin University. The chance to work as part of an international research community was a great experience for me.

I also want to thank my family, especially my father, for their support during my studies. It is very comforting to know that one can count on unquestionable aid and encouragement.

Finally I would like to thank my former girlfriend Vanessa for her support and understanding while I was occupied with my thesis project. It is not granted that a partner supports the decision to be separated for a long time.

Contents

1. Introduction

Recently, the overall assessment workload in education has increased greatly over all educational levels. As business structures changed, the need for more qualified workers increased, leading to a larger number of higher education students (Bruer, 1994). Associated with the increasing workload is the rise of overall costs of grading. Since the workload and time spent on evaluation increased, the time available for individual feedback decreased (Carter et al., 2003), contrasting new educational paradigms suggesting that suggest to give greater, continuous and individualised feedback to students. Based on constructivist learning theories more continuous assessment to support learning and teaching should be done at the classroom level (Shepard, 2000). Along with the structural and theoretical changes, the usage of computers in education has increased greatly. Computers are no longer used simply as tools for specific tasks. With the spread of computer networks, whole learning environments have been designed for local and distance learning. E-learning uses electronic student assignments, generating a need for electronic assessment as well (Bull & McKenna, 2004; Valenti, Neri, & Cucchiarelli, 2003). These changes in the educational environment raised the motivation to research into assistance technologies and automation of routine tasks.

Condensed, the drive for computerized assistance in assessment is build up by two different main factors:

- large class sizes, standardized tests and the associated costs for grading
- individualized feedback to improve learning

In both cases feedback must be given quickly after any assessment task.

Out of the initiated research, solutions for automated assessment in different forms, ranging from assistance to fully automatic systems, have been created for subjects as computer programming and mathematics. Essays are a universal question type for assesssment that is applicable to a wide range of subjects. The advantage of this question type, beyond the usage in writing instruction, is the possibility of testing for high-order skills.

1.1. Motivation

Designing tests requires the matching of learning objectives with adequate questions types. Bloom's taxonomy in the cognitive domain classifies and ranks educational objectives with matching skills and assessment actions (Bloom, 1956). Essay questions are the simplest type covering all six levels of Bloom's taxonomy and are especially suitable to assess at the synthesis and evaluation level (Valenti et al., 2003; *Exam Question Types*

and Student Competencies, n.d.). For this reason essays are widely used in assessment and therefore there is high interest in efficient and reliable essays grading.

According to Bereiter (2003) the major part of research funding on Automated Essay Grading (AEG) is related to mass testing of writing skills, producing a great number of essays or similar texts. Essays are produced under the same conditions to the same prompt. Existing solutions are therefore mostly applicable to large class sizes and require considerable training effort.

This makes AEG solutions unusable at the classroom level with low student numbers. Small class numbers are not the only major obstacle for existing solutions deterring the use of current systems at this level. In class, the provided feedback is extremely important which is not the case in mass testing, as students will often receive only the final score. Many existing solutions were primarily designed for this summative assessment application and therefore do not deliver rich and reliable feedback to students. A higher importance of feedback at the classroom level goes along with the interest a more formative assessment.

An essay grading solution covering both aspects of small class sizes and formative assessment could be integrated into many e-learning systems, greatly increasing the usefulness of the systems. As student numbers are much lower, such a solution does not necessarily need to be fully automatic. A semi-automatic approach assisting teachers by decreasing the time spent on grading would be of great benefit in this application.

A semi-automatic approach to essay grading also addresses objections against fully automated system that the teachers might have. As the final decision about grades is left to them, any objection questioning the reliability of the system is nullified. Teachers are motivated to adopt the system as the semi-automatic results decrease their workload of assessing submissions.

Another advantage of the semi-automatic approach is that teachers can give personal feedback to students. The full potential is acquired when the system actively supports teachers in giving feedback by providing analsis data which can be quickly transformed to personal comments by teachers. This avoids the problem of fully auotmaically generated feedback being preceived as formulaic by students (Chen & Cheng, 2008).

Providing detailed feedback to students and while speeding up grading at the same time is a challenging task. Analytic feedback inherently contains greater detail but is more demanding to teachers while grading. Some grading tools support analytic feedback better. Analytical rubrics define the criteria and achievable levels explicitly, easing the grading for teachers. Since the level definitions describe the expected achievements they are already a form of analytical feedback. A semi-automatic system can support teachers further by providing feedback suggestions to speed up the writing of comments to students.

The application at classroom level and the resulting benefits of a semi-automatic approach make such an essay assessment system a valuable contribution to the education community.

1.2. Structure of the Thesis

The remainder of this thesis is organized as follows: Chapter 2 discusses the theoretical background of the learning process. It continues with assessment, which is a vital part of the learning process and leads over to computerized assessment. Essays are introduced as an assessment type to test for expected skills, followed by a thorough explanation of rubrics as an appropriate essay assessment tool.

Chapter 3 describes state of the art methods of computerized assessment systems currently available. Focus lies on essay assessment and the availability of automatic features. Electronic tools for rubric creation and handling are also included in the review. The systems are analysed for opportunities and weaknesses in the summary of the chapter.

Chapter 4 outlines the design approach leading to a rubric-based essay grading system. Key issues for such a system are discussed and high-level requirements are specified. The chapter ends with a description of the proposed system.

In Chapter 5, the system is specified in further detail. To implement and evaluate the approach, the requirements for a first prototype are given. The architecture of the prototype is outlined and the implemented evaluation algorithms are described in detail. Used libraries and external services are included in the descriptions.

The thesis continues in Chapter 6 with a presentation of the actual implementation through screenshots and usage instructions. An example use case is used through to demonstrate and evaluate the capabilities of the prototype. Additionally, each implemented criterion is evaluated on its own.

The lessons learned during the whole project are stated in Chapter 7, covering the literature review, implementation and system usage.

Finally, Chapter 8 gives a summary and emphasizes future work and possible further usages of the system beyond the orginally targeted application area.

2. Background

To know how people learn is important for everyone involved in educational processes in order to create an efficient learning environment. Weimer (2002) has criticized that the higher education community had "*ignored learning*" for a long time as "*most assumed learning was an automatic, inevitable outcome of good teaching*" and therefore had fostered teaching skills reflected by a large volume of corresponding publications in the context of education in contrast to a much smaller volume dealing with learning (p. XI).

Bruer (1994) has contended that the expectations on the skills and knowledge of students have changed greatly. Many students can "*remember facts, solve routine textbook problems and apply formulas*" but few can use their knowledge to solve more complex problems (p. 5). Furthermore the expectations on comprehension, communication and reasoning skills of many entry-level workers have increased due to changed business structures.

The amount of available knowledge has increased greatly and so it is impossible to teach everything. As a result, the idea of life-long learning has been fostered by many researchers. As Weimer (2002) has pointed out, "*students need to graduate from college knowing as much about learning content as they know about the content itself*" and need to be able to "*relearn old information*" due to the "*evolution of knowledge*" (p. 49). Consequently, the goals in curricula and teacher instructions need to move away from teaching bits of knowledge for memorization and closer to fostering deeper understanding (Ramsden, 1992) and skills necessary for self-guided learning.

New findings in psychology and neuroscience change and extend our knowledge on how learning occurs in the human brain. Consequently, the educational system needs to be adapted to utilize the new findings to create more efficient learning environments and meet the changed expectations of more graduating students with high-order skills (Dochy, Gijbels, & Segers, 2006).

Traditionally, assessment through formal tests has been a part of the learning process to determine grades in most school systems. Changing educational theories have lead to a wider view of assessment. Assessment can directly serve the learning process of students as it gives valuable feedback about their current knowledge and directs further learning. In the following sections, the learning process is first discussed in more detail, followed by assessment in general and finally leading to essay assessment and related topics.

2.1. Learning Process

Advances in scientific fields such as developments in psychology and neuroscience provide fundamental understanding on different key aspects of learning. These research findings need to be evaluated in real educational settings to determine effective learning environments and discover influences through classroom interaction to the underlying theories. Current curricula still often focus on knowledge of bits of facts and the capability to depict this knowledge rather than on thinking critically and complex problem solving capabilities. Dewey (1997) defined the main goal of traditional education as transmission of the subject-matter worked out in the past "*consisting of bodies of information and skills*" to the new generation in an institutional environment bound by clear rules expecting conformity of students (p. 17). The author concluded that learning thereby means "*acquisition of what is incorporated in books and in the heads of elders*" in assumption that knowledge is static and therefore "*is taught as a finished product, with little regard to the ways it was originally build up or to changes that will surely occur in the future*" (p. 19). Concluding, the traditional education approach can be described as greatly teacher-centered as instructional design guidelines and methods focus on the knowledge transfer from teachers to the students.

The development of cognitive science as a multidisciplinary perspective has changed the way research on learning is conducted and produced a new understanding of learning. Bransford and Brown (2000) have explored the research work summarizing key findings and outlined recommended changes to educational processes on the basis of the reviewed research literature. They have reported three key findings concerning learning:

1. If the initial understanding that students bring to the classroom is not considered, they will fail to grasp the new concepts and only memorize information to reproduce it at a test (Bransford & Brown, 2000, p. 14,15).

2. Competence in a field needs "*a deep foundation of knowledge*", understanding of "*facts and ideas in the context of a conceptual framework*" and knowledge organisation that "*facilitates retrieval and application*" (Bransford & Brown, 2000, p. 16)

3. "*A metacognitive approach to instruction can help students learn to take control of their own learning by defining goals and monitoring their progress in achieving them.*" (Bransford & Brown, 2000, p. 18)

Considering the reason to learn the institutional educational process where basic education is mandatory in many countries can be differentiated from unsupervised, self-guided and self-motivated learning. For the latter Mitra et al. (2005) reported an experiment where children gain computing literacy skills on their own by self-motivated learning and suggests to apply this "*minimally invasive education*" in those circumstances where "*schools and teachers are either absent or not effective due to any reason*" (p. 1). This experiment is a further indicator that changes in teaching routines in schools and higher education could lead to better results. In contrast to the teacher centered approach in

traditional schools, Weimer (2002) has outlined a practical approach which is learner centered to move "*from passive dependent learners to autonomous, intrinsically motivated, and self regulating*" students (p. XX). The teacher's role is seen as a guide, facilitator and designer of learner experience, sharing the responsibility for learning with the student. The underling view of this practical approach has been described by Bransford and Brown (2000, p. 133) with the four partly overlapping perspectives on learning environments: learner centered, knowledge centered, assessment centered and community overlapping the previous three. For learning environments to be most effective, all perspectives should be addressed in the instructional design. Learner centered environments take the previous knowledge, beliefs, understanding, motivations and cultural practice of students into account. "*Well organized bodies of knowledge*" are the basis to support more general problem solving strategies which is reflected in the knowledge entered perspective (Bransford & Brown, 2000, p. 136). The necessary coverage of knowledge is traditionally addressed in curricula. The key aspect in a modern implementation is to avoid the listing of isolated objectives but promote deeper understanding of conceptual ideas and therefore the connection between the objectives. The perspective of community in learning environments not only comprises the classroom and school level and the social norms necessary for learning there but should be seen as "*connected to the larger community of homes, businesses, states, the nation, and even the world*" (Bransford & Brown, 2000, p. 145). This is partly due to the expectations and goals a society has in its educational system and as well to the fact that students who are given a reason why they learn are more motivated. A student who does not see the benefit and application of the learned knowledge in real life contexts will mainly try to pass the tests with a memorizing learning strategy and not gain the understanding to transfer the knowledge to new domains and problems. The assessment perspective is important especially when considering the goal of learning with understanding rather than memory for facts and procedures. This is linked with the third key finding of a metacognitive approach to learning. To enable students to monitor their own progress they need to know their own knowledge state. What a student has learned is exactly what assessment traditionally tries to determine. This form of assessment is also called *assessment of learning* (L. M. Earl, 2003; Birenbaum et al., 2006). For a student to become metacognitive this form of assessment is not sufficient as it usually occurs only at specific points in time usually mainly focusing on the goals of grading and ranking. To enable students to monitor their progress and adapt their learning more frequent assessment is necessary. This kind of assessment is often called *assessment for learning* (L. M. Earl, 2003; Birenbaum et al., 2006; Shepard, 2000) and can be utilized by teachers to adapt their instruction and give richer feedback to students. If additionally students also assess themselves they are becoming metacognitive about their own learning. Consolidated assessment should be seen as an integral part of the learning process and will be discussed further in the next section.

2.2. Assessment

Assessment in education documents knowledge, skills, attitudes and beliefs using some kind of measurement. How and when assessment is done depends on the learning objective which is linked to the specific learning or eduction theory used in instruction. Individual students, the learning community, the institution or the educational system as a whole can be the subject of assessment. The former two are often called classroom assessment while the latter two are usually large-scale assessments. Aside from the obvious reason to determine grades assessment is carried out to determine the effectiveness of instruction, to motivate learners, to provide feedback to learners about their current knowledge level and to gather data for statistical analysis to name a few other. These reasons stand behind the goals of classroom and large-scale assessment but are weighted differently in these two contexts. Different goals can lead to different types of assessment as each type may foster certain goals better than others or conflict with one another (L. M. Earl, 2003). Pellegrino, Chudowsky, and Glaser (2001) have reported that "*often a single assessment is used for multiple purposes*" but "*the more purposes a single assessment aims to serve, the more each purpose will be compromised*" (p. 2).

R. James, McInnis, and Devlin (2002) have argued that assessment should be seen as a strategic tool to enhance teaching and learning due to the fact that students often "*work backwards through the curriculum, focusing first and foremost on how they will be assessed and what they will be required to demonstrate they have learned*" (p. 8). This student view of assessment is based on the grade determining function of assessment. For the teacher this results in the requirement to consider the effects of assessment during the planing of a course. The real progress of students cannot be foretold reliably during the planning of instruction. Therefore the assessment done during a course should be used to revise and adapt the plan. It is important that teaching and instructions are adapted but to ensure the educational expectations of what students are intended to learn stay unambiguous to them.

Any assessment is a form of feedback to students at least in the sense to determine their current performance level. Shepard (2000) has pointed out that the view on feedback differs greatly between behaviourist assumptions and constructivist perspectives. Despite existing differences it is considered an important factor in learning in both theoretical directions. Feedback is linked to self-correction and improvement by the learner himself but is an external input usually provided by teachers or peers especially in group working tasks. Bransford and Brown (2000) have reported that to gather expertise in a subject one needs, besides other skills and capabilities, to "*become metacognitive about their learning so they can assess their own progress and continually identify and pursue new learning goals*" (p. 50). This contrasts to feedback in the aspect that the state of ones current progress is determined by oneself. Therefore the capability to perform self-assessment, interpret and use the results to achieve improvement is an important part of being metacognitive.

The type of assessment influences the learning methods students use as well as what and how much they learn (Gijbels & Dochy, 2006; Ramsden, 1992). Choosing the appropriate method requires a great deal of knowledge about the different methods. For

example Knight (2001) listed around 50 assessment techniques to outline the multiple options available. It is out of scope to go into details of every technique. Instead the next sections outlines common criteria to differentiate various assessment techniques.

formative - summative

R. Stiggins (2007) has stated that the "*major role of assessment has been to detect and highlight differences in student learning in order to rank students according to their achievement*". This type of assessment is often called *assessment of learning* or *summative assessment*. It is typically performed at the end of a period like a course or semester to determine the outcome of the learning period and derive the grades for the students. R. J. Stiggins, Arter, Chappuis, and Chappuis (2004) have argued that standardized testing and determining grades is assessment done solely for accountability reasons and therefore does not benefit learning.

In contrast to this the more modern view of *assessment for learning* exists, also called *formative assessment*. It is not necessarily bound to a specific point in time and can be applied continuously. Its purpose is to provide feedback about the current state of knowledge to both students and teachers to adapt teaching and learning activities. Formative assessment can be initiated and carried out by teachers, peers or by individual learners as self-assessment. P. Black and Wiliam (1998) have considered assessment to be formative only if the gathered "*evidence is actually used to adapt the teaching work to meet the needs*". This emphasizes the different purpose of summative and formative assessment. Applied to summative tests conducted during the semester certain tests can turn into formative assessment if the results are used to adapt the teaching process for the rest of the semester (Atkin, Black, & Coffey, 2001).

objective - subjective

Objective assessment is a form of testing where each question has a single correct answer which is unambiguous and can be "*marked without any judgement made on the part of the marker*" (Freeman & Lewis, 1998, p. 78). Examples are true/false answers, multiple choice and matching questions. Mathematical problems which have a single final result can be seen as a special case of objective questions. Although the approach of solving the problem may vary there is an unambiguous final result. Subjective assessment is a form of questioning with a variety of correct answers. Free form answers are ambiguous as the same fact can be expressed in different ways. Essays clearly match this definition as they allow the greatest variation in answering a prompt. Subjectiveness is not only present in the answer text but in grading as well resulting in discrepancy in ratings by multiple judges. The concordance between two or more essay judges is described as inter-rater reliability using different methods (Gwet, 2001; Blood & Spratt, 2007; McGraw & Wong, 1996; Wikipedia, 2010b).

informal - formal

Formal assessment occurs in a structured way often utilizing a written document contributing to the final grade. Standardized assessment falls into this category as the assessment is formally specified in every aspect ranging from questioning to evaluation. Informal assessment is more casual and usually carried out while interacting with the class. R. J. Stiggins et al. (2004, p. 93) has defined informal assessment as any teacher student interaction revealing what students have learned without recording of student answers. The author gives the following examples:

- questions asked during instruction
- examining student participation in class
- reading and responding to students' comments in journals and logs
- oral examinations
- student interviews in conferences

Informal assessment is often related to formative assessment but formal assessment can be used in formative ways as well (Atkin et al., 2001; McMillan, 2000). Another indication for informal assessment is that students may not recognize it as assessment at all (Atkin et al., 2001; Brualdi, 1998). E-learning systems which adapt the course structure and content to specific learners must determine their level of knowledge. The necessary assessment can be considered formative as the content adaptation is analogous to teachers adapting their instructions and informal as the user may not notice the continuous assessment. An example for this is AdeLE which uses eye tracking and other user behavioural traits as for example frequency of visits or time spent on learning objects to build a learner profile and adapt the presented learning content to the user (Gütl et al., 2005).

primary trait - holistic - analytical

Depending on the objective of assessment the result can be a measure of a single primary trait, holistic or analytical[1](Liz, 2003, p. 175). In a holistic score several features are considered but only one single rating comprising all features is given. In the context of grading essays holistic rating corresponds with the view that a reader of an essay is naturally not evaluating mechanics, style and other features distinctively but rather gets an overall impression resulting in an evaluation of the essay.

Table 2.1.: Scales for writing assessment

	specific to a task	generalizable to a class of tasks
single score	primary trait	holistic
multiple scores		analytic

(Weigle, 2002)

[1]some authors use the term multiple trait scoring when different traits are assessed distinctly

Similarly primary trait scoring results in a single score as well as only one trait is considered in assessing. Primary trait scoring is applicable only to a specific task or context while a holistic scale can be used for several similar tasks. Although only one trait is of interest still several features contributing to the trait in question may be evaluated while others may be neglected entirely (McTighe & Arter, 2004). Neglecting features not relevant to the primary trait may be difficult for raters and must be considered in test design and training (Saunders, 1991).

Analytical assessment evaluates each feature separately thus resulting in multiple scores. Depending on the objective of the assessment task representative features will be selected for evaluation and irrelevant ones neglected. Weigle (2002) has summarized the differences between the three scoring types for writing assessment in Table 2.1.

In conclusion assessment should be considered as an integral part of the learning process regardless of the theoretical framework on which the educational settings are based. Besides the different requirements out of the frameworks to perform assessment it is often prescribed by local, district or state agencies in specific forms and frequency mainly for accountability reasons. As outlined many different forms of assessment exist and the type of assessment influences the learning methods students use as well as what and how much they learn (Gijbels & Dochy, 2006; Ramsden, 1992).

Reliability in scores is an important issue in assessment and especially crucial in high-stakes assessment. Bias and subjectivity of raters can influence student scores massively. Marking a huge amount of assignments is tiresome and rater performance will vary over time. Marking is a cost factor as it needs time and possibly additional staff. It can also be argued that time might be better spend on teaching in many cases. Computerized assessment is addressing these issues. Human raters can be at least partly replaced by computers saving time and possibly money. Well designed scoring algorithms are not prone to the bias related to social background, handwriting and similar aspects as human raters might be. The next section first outlines motivations for computerized assessment and different types of it are defined. It is followed by a discussion of computerized grading and finishes with validity considerations of computerized assessment.

2.3. Computerized Assessment

Increasingly large class sizes and a limited amount of time for assessment have raised the interest in computerized assessment to reduce the costs associated with grading and the lag between test administration and test reporting (Chung & O'Neil, 1997; Palmer, Williams, & Dreher, 2002). Another factor especially in summative assessment is the objective to ensure reliability (sometimes called consistency[1]) so that work of the same quality receives the same mark. Considerable effort is needed to achieve this in the case when multiple human graders are needed due to the large amount of students. Even in the case of a single grader the performance can vary over time resulting in variation of the marks. Computerized systems can as well offer opportunities for formative assessment

[1]Lang and Wilkerson (2008) discusses reliability vs. consistency

by providing instant feedback to students which may result in more widely acceptance of automated assessment systems (Shermis & Burstein, 2003). A formative e-learning system should continuously assess the students and provide immediate feedback to them to revise their work or adapt their learning strategy (Gütl, 2008a). This does not only apply to e-learning systems but to all open-ended learning settings. In e-learning system uses can be continuously assessed which can potentially occur unnoticed to the user to adapt the learning content. This can only be achieved effectively with computerized approaches in assessment. Different application domains for computerized assessment as programming classes and mathematics have lead to dispersed and distinct solutions serving the special domain needs (AL-Smadi, Gutl, & Kannan, 2010). While this solutions deliver the advantages in their domain more general and flexible solutions can be of far greater value especially in emerging new learning settings as described by Chang and Gütl (2010).

Semi-Automatic and Fully Automatic Assessment

Computers can be utilized in two different ways for assessment. First they can be used as an aid for human raters manually grading a submission. This can be either done by a partial automatic evaluation of the submitted works providing part of the final score were the teachers does the final grading or as a mere assistance during manual grading as for example by highlighting spelling errors in essay grading. The second way is to fully automatically evaluate submissions. In this case teachers do not grade each submission individually any more. The first is usually called computer-assisted assessment (sometimes aided) though the term is not unambiguous as it is used for fully automatic assessment systems and partial automatic system as well (Winters & Payne, 2005; Sim, Holifield, & Brown, 2004). To differentiate fully automatic systems from partially automated systems the term semi-automatic is used by some authors (Shortis & Burrows, 2009; Ala-Mutka & Järvinen, 2004).

The definition of semi-automatic systems by Shortis and Burrows (2009) focuses more on overall teachers assistance like feedback phrases than on automatic analysis at all. In the context of computer programming classes Ala-Mutka and Järvinen (2004) defined semi-automatic assessment as a fully automatic evaluation by testing the program for specification fulfilment and an additional manual grading and feedback process covering complex issues not evaluated by the automatic testing. A fully automatic system can still require considerable human effort as systems typically need training data to build their scoring algorithm. (Kakkonen, Myller, & Sutinen, 2004) have used a slightly different definition not focusing on human involvement. In their definition a fully automatic system gives only a score (summative assessment) while a semi-automatic provides a grade and more detailed feedback to support learning (formative feedback).

Computerized Grading

Objective type questions as multiple choice can easily be evaluated automatically (Bull & McKenna, 2004) without doubt on reliability or bias. E-learning management systems have support various support for different objective type questions. Research in this area is more related on the utilization and implications of such system rather than on the technical background. As the gains in efficiency are much higher for computerized essay assessment compared to objective assessment forms substantial research is going on in the area of AEG. Automated Essay Scoring (AES) is used by some authors as an alternative name and a few refer to it as Automated Writing Evaluation (AWE). The last emphasises that a system can be applied to texts besides essays and may provide more feedback than a single score (Warschauer & Ware, 2006). AEG systems can by design either be semi-automatic or fully automatic. Fully automatic systems provide the fastest feedback to students and the biggest cost advantages[1]. Semi-automatic systems usually only provide automatic analysis and maybe suggestions to human markers who in the end decide about the actual score.

Validity

Validity can be easily proven for evaluation of objective type questions but it an issue in any subjective assessment type. Chung and O'Neil (1997) have argued the assumption "*human ratings are the best estimate of the true score*" is the basis to assess the performance of an AEG system by measuring the correlation between the scores. Stemler (2004) has argued that expressing inter-rater reliability in a single value like a correlation coefficients is imprecise. As a literature review such as Gütl (2008b); Ben-Simon and Bennett (2007); Williams and Dreher (2004); R. James et al. (2002) reveals this kind of measurement is the main criterion used to demonstrate validity of AEG systems. Often high correlation values have been reported for later commercialized systems which support the use of the systems for mass grading. Wang and Brown (2007) have contended that "*very few studies have been conducted by independent researchers and users*". Wang and Brown reported a significant difference in the performance of IntelliMetric™ compared to previous studies related to the developers of the system.

In high-stakes assessment a hybrid scoring approach may be used to address the issue of validity and avoid legal concerns about automated scoring. For example in the Graduate Management Admission Test (GMAT) essays are once automatically assessed and once rated by one human rater to discover discrepancies. In case of a too large difference between the automatic and the human score the essay is rated by a second human rater (Burstein, 2003; *GMAT® Scores and Score Reports*, n.d.). Burstein has reported that only in 3% of the cases the score discrepancy was 2 or more points requiring a second human rater. C. L. James (2006) has contended that by combining the scores by IntelliMetric™ and "untrained" human raters into one logistical regression model the prediction success of student class placement slightly increased. This result is an indication that the hybrid approach is a good use case of current AEG systems. This

[1] cost advantages can be reduced through running license costs in commercial AEG solutions

is backed by the result of a study by Powers, Burstein, Chodorow, Fowles, and Kukich (2002) where several challengers intentionally tried to fool the e-rater scoring engine. The study reported that with knowledge about the engine it was possible to successfully fool the engine. For high-stakes assessments the authors therefore have contended not to use an AEG system as the sole method of scoring.

The correlation criterion for validity is not sufficient for systems used in formative assessment. Kakkonen and Sutinen (2008) have proposed a wider set of criteria for evaluation of AEG systems but do not define precise metrics and measure for all of them. Concluding from the differentiation between holistic and analytical scoring (see 2.2) it seems desirable to have an analytical rating which provides more feedback to students. Valid and rich feedback enables students to realize their own progress and gives hints on how to improve their achievement.

Numerous assessment methods and prompts exist including written answers, oral exams, presentations and other forms. Mainly written assignment types can be computerized to various degree. Which method is used should mainly depend on the educational and assessment goals but choice may be limited by available resources for assessment. Considering Bloom's taxonomy essay questions are a relatively simple form covering all six taxonomy levels including synthesis and evaluation level reflecting higher order skills expected in graduating students (Bloom, 1956; Valenti et al., 2003; *Exam Question Types and Student Competencies*, n.d.). According to Scouller (1997) students with a positive perception of essay assessment did so because it "allowed them to develop higher order intellectual skills and abilities". Therefore essays are a viable assessment method both from a theoretical viewpoint as well as students perception of assessment. As a written question type the possibility to computerize essay assessment is given when they are submitted in digital form. The next section will discuss essay assessment in more detail.

2.4. Essays

Objective type questions as multiple choice, short answer, selection/association and similar can be assessed faster and may therefore be preferred under time constraints by teachers. According to Valenti et al. (2003) researchers do agree that certain aspects of complex achievement cannot be easily assessed through this test item. Bloom (1956) first published a taxonomy for the cognitive domain classifying and ranking educational objectives with matching assessment actions. Selected-response assessments as multiple-choice, matching and true-false tests do not require certain skills as for example composition and expression. Assessment types forcing students to develop their own answer demand high-order skills. This type of assessment is called constructed-response. Response types range from written texts as essays to live performances and work products (Stecher, Rahn, Ruby, Alt, & Robyn, 1997, p. 23-24). Essays cover all six levels of Bloom's taxonomy and are especially suitable to assess at the synthesis and evaluation level[1] (Valenti et al., 2003; *Exam Question Types and Student Competencies*, n.d.;

[1] equalling the levels evaluating and creating in the Revised Bloom's Taxonomy (Forehand, 2005)

Bloom, 1956). According to Richardson (2002) essay questions should be only used when other types of assessment questions cannot measure the desired objectives. They should be used to test higher order skills rather than basic skills as memorization and recalling.

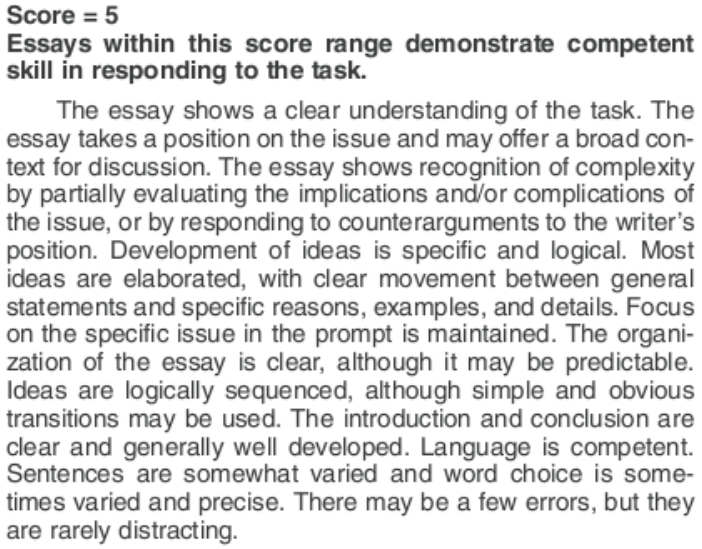
Grading essays is more challenging than grading objective types questions. Besides that more time is needed to grade essays compared to objective type questions there is also a higher possibility of bias, subjectivity and unreliability in scoring essays (Richardson, 2002). Bias can be based on students opposing position to the scorer's position on a controversial subject. The order of marking essays can influence scores as graders remember previously read essays and may become more lenient or critical. Tiredness can affect judgement as well. Essay are prone to the halo effect that excellent writing skills can lead graders to become lenient on shallow content in an essay (Bereiter, 2003). These problems can be partly addressed through proper teacher training and the development of scoring guidelines (Wang & Brown, 2007; Weigle, 1999). Training human raters costs time and money and might not always yield the desired effect especially with seasonal staff (Barrett, 2001). Without a proper scoring guideline any rater training will intrinsically fail. A good scoring guideline should be understandable both by teachers and students and must be appropriate for the assessment task. Especially in a formative setting it is important that students understand why they got a certain grade on their work. It is therefore often recommendable that assessment criteria are communicated before the actual assignment takes place. For essays rubrics are a common scoring tool used to achieve these goals and will be discussed in detail in the next section.

2.5. Rubrics

The grade a student receives for an assignment depends not only on the student and his/her achievement but as well on the rater and the applied rating scale. Rubrics are tools for assessment appropriate to subjective assessment tasks (see Chapter 2.2) as they specify what is measured in a descriptive way allowing to judge the quality of a work. Moskal (2000) has maintained that the subjectivity in essay grading becomes more objective due to the predefined evaluation scheme comprised by a rubric.

A rubric can be defined as a list of scoring guidelines (category scales, criteria) each describing different levels of performance by defining the characteristics for distinct levels (Andrade, 2000). Rubrics can either be holistic or analytical. In case of a holistic rubric a single scale with several descriptive levels exists. The definition of a level can comprise several distinct features considered to be in line with each other (Arter & McTighe, 2001, p. 18).

In an analytical rubric several criteria are evaluated and each criterion has a separate scale (Brookhart, 1999). The separate scales can be weighted differently according to the emphasis of the represented criteria. The number of levels can differ between the single criteria. Brookhart (1999) has recommended "*to use as many levels as there are meaningful distinctions*" (p. 46). A rubric can be designed for a specific task or generally for a class of similar tasks. A general rubric is limited as it cannot cover, for example, content aspects specific to an assignment. Table 2.3 summarizes the options of analytical



Score = 5
Essays within this score range demonstrate competent skill in responding to the task.

The essay shows a clear understanding of the task. The essay takes a position on the issue and may offer a broad context for discussion. The essay shows recognition of complexity by partially evaluating the implications and/or complications of the issue, or by responding to counterarguments to the writer's position. Development of ideas is specific and logical. Most ideas are elaborated, with clear movement between general statements and specific reasons, examples, and details. Focus on the specific issue in the prompt is maintained. The organization of the essay is clear, although it may be predictable. Ideas are logically sequenced, although simple and obvious transitions may be used. The introduction and conclusion are clear and generally well developed. Language is competent. Sentences are somewhat varied and word choice is sometimes varied and precise. There may be a few errors, but they are rarely distracting.

Figure 2.1.: Level description of the ACT holistic writing scoring rubric (ACT, 2009, Appendix I)

and holistic rubrics.

The format of a rubric is very flexible and only bound by constraints like for example readability, clarity and overview. Mostly list or table representations are used (See Figure 2.2 for schematic layout examples). For analytical rubrics often the table layout is preferred as it ensures a good overview over the different criteria and levels used. If the description of each level is too exhaustive in length table cells might get too large and a list layout may be the better choice. In both formats the levels are sorted but no consensus exits whether to begin with the highest or lowest level. Each level description should give a clear indication of the expected performance to reach the level. It is commonly considered useful to include short examples especially in cases where they can replace a lengthy explanation. As a rubric is designed for specific assignments or a class of similar tasks the application context needs to be specified. This can be done in both formats in a preface which might include a title for the rubric.

Numerical scores It is rather straight forward to get a numerical score from a holistic rubric. For analytical rubrics several possibilities to calculate a final overall score exist. Each criteria can be weighted differently or all criteria can be considered equally important. The same differentiation is possible for the scales where each level can have the same weight or be weighted distinctly. For example teachers may prefer to weight the lowest level differently to express the minimum achievement required to pass. The points for each criteria are summed up for the final score. According to Trice (as cited in Mertler (2001)) rubric levels should not be seen as percentages as more levels at and above average than below average performance may be defined in the rubric. A specific conversion from rubric points to grades or percentages should be defined explicitly to address the problem of non-linear scales within a rubric.

Level descriptions The level descriptions are the most important feature of a rubric to communicate the performance expectations and to facilitate appropriate scoring. Rubrics cannot only be used for scoring purposes but as mere instructional tools as well. The four listed level descriptions for the exemplary criterion "the claim" of an analytical rubric are taken from (Andrade, 2000).

The selected rubric is designed to help students to write persuasive essays:

1. I don't say what my argument or claim is.
2. My claim is buried, confused, and/or unclear.
3. I make a claim but don't explain why it is controversial.
4. I make a claim and explain why it is controversial. (Andrade, 2000)

In this example the expectations for each level are expressed in the view of the student. This is done as the rubric is intended to be used by students for self assessment while writing their essays to improve their performance. Andrade has suggested that this kind of rubric can be designed together with the students in the classroom. This promotes the students' understanding of the expected qualities of their work. Figure 2.1 is an example level description of the six-point holistic scoring rubric for the ACT Writing test (ACT, 2009, Appendix I). The example shows that level descriptions of holistic rubrics tend to be more exhaustive to clearly express what is expected for each level. Figure 2.2 is a partial example of an analytical rubric in grid form (Allen & Tanner, 2006). In this rubric for each criterion 3 levels exists. Instead of numbering them the points associated with each level are used as column headers and thus all criteria are considered equally important. These three examples indicate that designing a good and valid rubric takes time and is not an easy task.

Criteria Rubrics are applicable to a great range of assignment tasks like for example oral examinations, presentations, practical works, interviews and essays. The criteria used in such rubrics differ greatly but may be grouped into classes for evidence related to content (knowledge), construct (reasoning process), argumentation, skill competence, style or student progress. The tested "criteria should be directly related to the knowledge, critical thinking or skills" which students should acquire in a course (Brookhart, 1999, p. 46). Levels should not be specified by a term as *"good"* but have to be descriptive as for example *"few spelling errors not influencing readability"*. In the context of essay writing grammar, usage (word choice), mechanics (spelling), style, organization (structure), discourse, content (knowledge, topics, ideas) and conventions (as citation, bibliographies, appropriate usage of figures, tables) are often used as broad criteria. Kakkonen and Sutinen (2008) list morphology, syntax, semantics and discourse as linguistically defined levels of language knowledge. Depending on the prompt other criteria like originality may be the most important neglecting the previously outlined common criteria at all. The analytical rubric intended to assess critical thinking in a written essay by P. Connors (2008) comprised by the criteria (i) "Investigative question" (ii) "Concise accurate

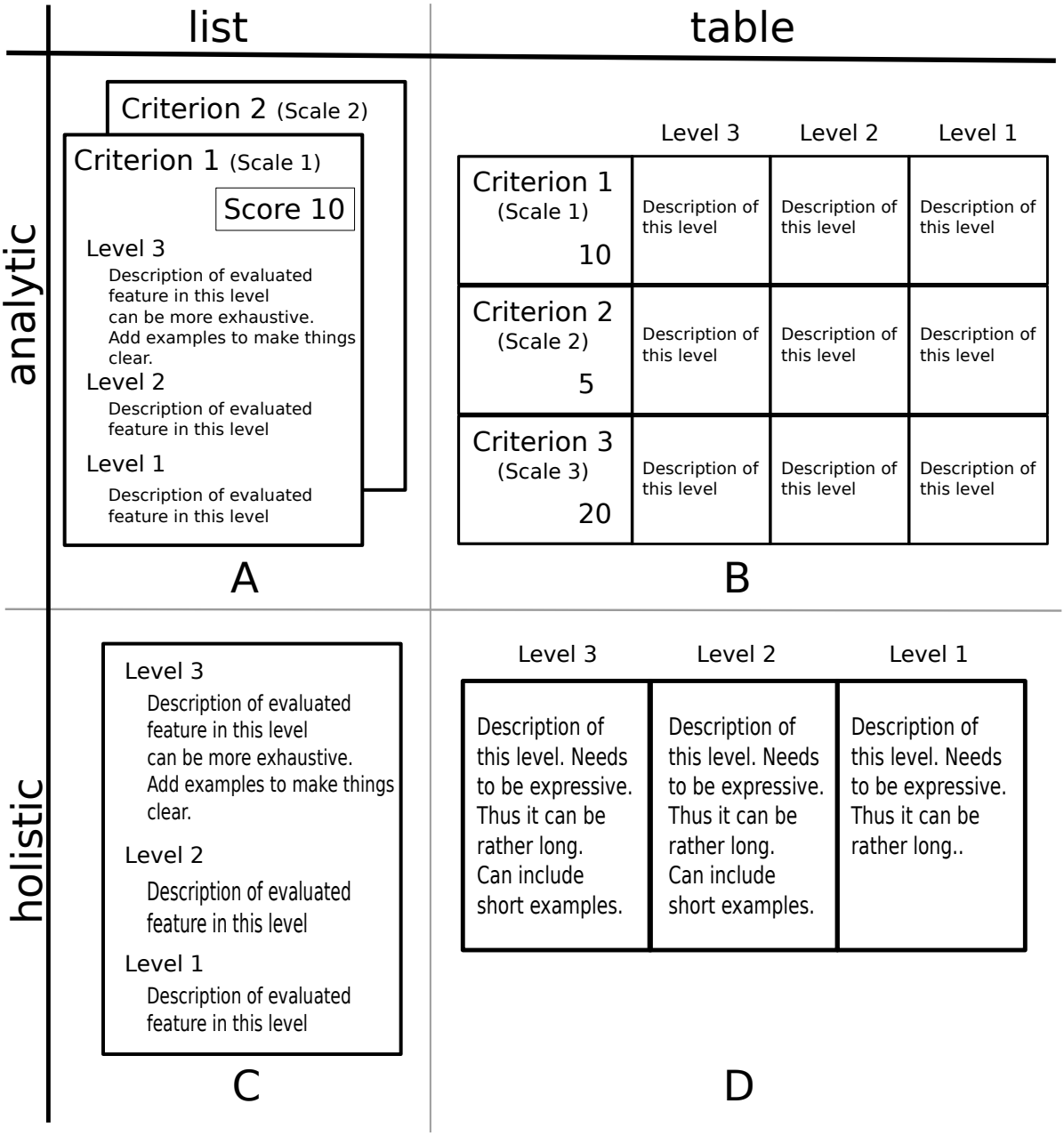


Figure 2.2.: A: analytical rubric, list layout, two criteria with explicit weight; B: analytical rubric, table layout, three criteria with differing weights; C: holistic rubric, list layout; D: holistic rubric, table layout

answer" (iii) "Samples from published research" (iv) "Confidence in conclusions" and (v) "Overall statement quality" is an example for a rubric for essay assessment not using any common criteria listed previously except for the requirement to cite published research.

Content and Concepts

Designing a rubric needs careful considerations of how student assignments are assessed. One criterion for validity of assessment instruments is content-related evidence (Moskal & Leydens, 2000). Although some specific essay prompts where writing mechanics are the focus do not require content evaluation this is rarely the case in higher education. Generally in higher education it is of greater interest which ideas an essays covers in respect to the body of knowledge from which a prompt is derived rather than mechanics. Direct textual comparison of essays to the knowledge body is problematic as language allows a great variety in expressing the same ideas. Therefore an abstraction mechanism for the ideas contained in the essay is needed. This abstraction is best described as concepts expressed in the essay which can be matched with the knowledge body. Although the exact notion of concepts is of philosophical dispute (Margolis & Laurence, 2006) resulting in five different main theories (D. Earl, 2007) they are regarded fundamental in cognitive science for the human mind (Margolis & Laurence, 1999) and so for learning. In the context of content evaluation of essays lexical concepts like "animal" corresponding to items in natural language (Margolis & Laurence, 1999) are the most basic practically usable definition. In this sense concepts stand for the semantic of words, sentences and paragraphs expressing the ideas covered in the essay.

Computationally discovering concepts in written text is a complex tasks. Williams (2007) has outlined the Normalised Word Vector approach to represent the content of essays for automatic grading purposes. For each relevant unit like words and phrases the semantics are represented by a core concept derived from a thesaurus. This greatly reduces the dimensions needed to represent the content and therefore allows faster computational processing. This is akin to Freges distinction of "sense" in the term of *mode of presentation* and "reference" (Margolis & Laurence, 1999). A referent can be seen as the concept or idea a certain language term is referring too. The fact that different terms or expressions can refer to the same referent is expressed in the notion of senses which characterize the referent and thus are a mode of presentation for it. This is analogous to the Normalized Word Vector algorithm where for each textual unit the thesaurus concept which it represents is determined for further processing.

In all cases where content is considered a vital part of essay writing this needs to be addressed somehow in the scale used for assessment. For example the coverage of required concepts could be expressed as a separate criterion in a grading rubric. Due to the fact that the same expression in language can refer to different concepts the context or knowledge domain should be defined further either by providing examples or reference material covering the expected knowledge body.

Table 2.2.: Two criteria and level descriptions of an analytical rubric

No.	Criteria	2 points	1 point	0 points
1	*Demonstrates an understanding that ...* Food can be thought of as carbon-rich molecules including sugars and starches.	• Defines food as sugars, carbon skeletons, or starches or glucose.	• Attempts to define food and examples of food, but does not include sugars, carbon skeletons, or starches. • Must go beyond use of word food.	• Does not address what could be meant by food or only talks about plants eating or absorbing dirt.
2	*Demonstrates an understanding that ...* Food is a source of energy for living things.	• Describes food as an energy source and discusses how living things use food.	• Discusses how living things may use food, but does not associate food with energy.	• Does not address the role of food.

(Allen & Tanner, 2006, Table 3)

Table 2.3.: Rubrics design possibilities

holistic	analytical
one scale consisting of 1 or more features	several independent scales (minimum two)
task specific or general	task specific or general
single score	several scores and overall score
no explicit weight for a single criterion	explicit and differing weights for criteria

2.6. Summary

Expectations on the skills of students have shifted from fact memorization and routine problem solving to more high-order skills and self-directed learning reflected in necessary changes in educational systems (Dochy et al., 2006; Weimer, 2002). Our knowledge about key aspects of learning is increasing stressing the importance of assessment as part of the learning process. Assessment can serve different purposes in educational systems ranging from accountability (effectiveness of instruction) and differentiation (grading, ranking, certifying) to feedback to support learning. This is expressed in the views of summative and formative assessment resulting in differentiated assessment methods. Formative assessment is generally supporting student learning and therefore important to be applied continuously.

Furthermore it is important that grading criteria are communicated effectively between teachers and student before the assessment task is carried out. Jonsson and Svingby (2007) have contended that for reliable scoring of complex tasks requiring judgement of quality as is the case in essay grading a topic specific analytical rubrics is the best choice. Learning and instruction can be improved as expectations and criteria are explicitly recorded in a rubric which promotes self-assessment and teacher feedback. Bresciani et al. (2009) reported that the usage of analytical rubrics can yield to high inter-rater reliability levels even among untrained raters across multiple disciplines. These findings support the usage of analytical rubrics especially when multiple raters are involved.

To choose between different test items it is necessary to consider the purpose of assessment and what exactly should be assessed. The essay question item is suitable for summative and formative assessment as well and can be used to test high-order skills as it is covering all six levels of Bloom's taxonomy (Valenti et al., 2003; *Exam Question Types and Student Competencies*, n.d.; Bloom, 1956). Grading essays is more challenging than objective types questions as not a single right answer exists and many factors can influence judgement. Computerized assessment offers opportunities to address the problems of human introduced bias and variation. A main driving force for computerized assessment is to timely assess student works especially in large scale testing with lower human effort resulting in monetary benefits. This is less important at class-level with low student numbers where semi-automatic assessment approaches can still deliver the same benefits in assisting teachers to free up at least part of the time spent on grading.

3. State of the Art Methods and Approaches of Computerized Assessment

This Chapter discusses currently available computerized assessment solutions and approaches in the context of essay assessment and rubrics. The review provides a basis to identify advantages delivered by current solutions as well as limitations of the approaches which both will be discussed after the systems reviews. Automated features assisting assessment are of special interest but also other computer-assisted systems are included in the review. To this end, a review of different types of computerized systems intended for evaluation of essays and similar free text responses is done and the underlying principles and feedback features are shortly outlined.

To classify the systems in regard to automatism is a bit ambiguous as several different terms are used by various authors. Computer-assisted assessment is particularly ambiguous as it is used for fully automatic assessment systems and partial automatic systems as well (Winters & Payne, 2005).The definition followed in this thesis is that any system utilizing a computer is considered computer-assisted. Kakkonen et al. (2004) classified essay grading systems in manual, semi-automatic and automatic systems using several aspects like automatism and especially feedback. Based on this differentiation the reviewed systems will be classified into the three groups "manual and/or computer-assisted", "semi-automatic" and "fully automatic" with slightly different definitions focusing mainly on the aspect of automatism in respect to access essays. Definitions are given at the beginning of following sections followed by the reviews matching the system.

3.1. Manual and/or Computer-Assisted

All paper based assessment systems are clearly manual. Therefore every software system simply mimicking and replacing a paper based solution is considered a manual approach. All administrative grade-book solutions fall into this category if the assessment of a single assignment is completely done by hand either offline or electronic. Simple automatism like calculation of final grades and grade delivery to students fit the definition as a manual computer-assisted approach.

As the overall interest lies in automated features pure electronic grade-book solutions have been skipped from this section although they would fit the definition. Instead solutions centred around or including rubrics have been selected for review. Quite a few systems found are software solutions merely for creating and managing rubrics to print

them out for manual offline usage.

3.1.1. RCampus and iRubric™

RCampus is a web based solution for managing classes including classwork and grades. It contains the iRubric™[1] tool to create rubrics which can be used to assess the classwork. The available functions are broad and enable teachers to manage handling of different types of assignments and all associated grades. Using a rubric for grading is optional for each assignment.

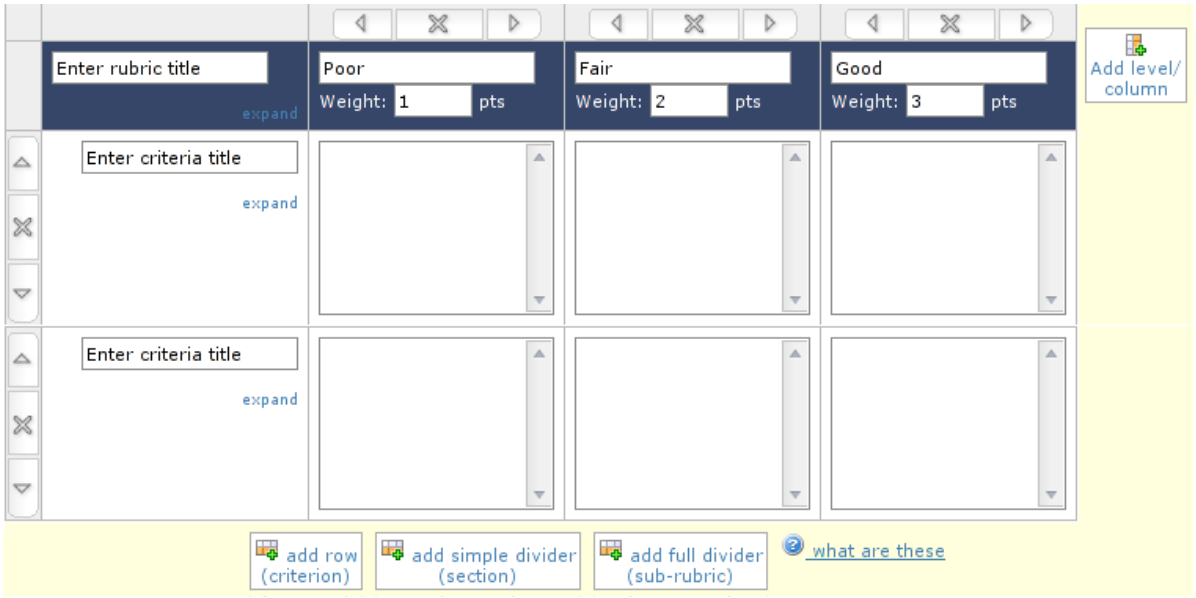


Figure 3.1.: iRubric™ Website Screenshot, rubric editing
(Reazon Systems, 2010)

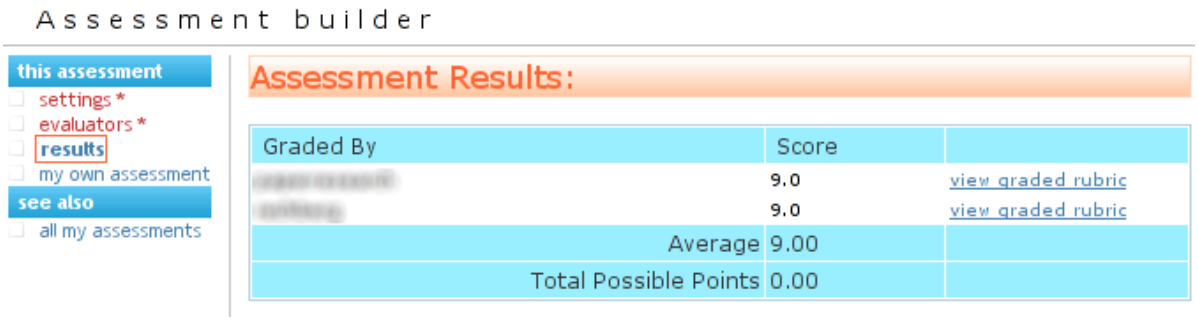


Figure 3.2.: iRubric™ Website Screenshot, scoring collaboration
(Reazon Systems, 2010)

The rubrics are edited and displayed in a grid layout which allows analytical rubrics (see Figures 3.1 and 3.3). Regardless of the actual criteria count a final single score is

[1] http://www.rcampus.com/indexrubric.cfm

calculated depending on the points associated to each level. To grade with the rubric the teacher simply clicks the achieved level. It is possible to add a comment to each criterion which students will be able to see if they are allowed to see their own grades. If students can see their grades online they also have the ability to write a message to the teacher to discuss their grade. No automated evaluation is done at all. Grading and textual feedback are completely done by the teacher.

Several statistics per assignment are available. Often no clear indication is given if a link or action will open a new browser window. The state of the previous and new window sometimes are related to each other and interaction is occurring. It is therefore possible to lose the overview as an untrained user.

The system provides community features for classes and teams for general discussions and document sharing. Similarly the rubric part offers to publish rubrics, discuss them, try them out and to evaluate arbitrary resources in a collaborative manner. The collaborative evaluation is not necessarily linked to a class or group. It therefore can be used completely independently by creating an assessment and inviting people to independently evaluate it utilizing the provided rubric. It is then possible to view the average ratings and to review the individual results (see Figure 3.2).

Figure 3.3.: iRubric^{TM} Website Screenshot, rubric feedback (Reazon Systems, 2010)

3.1.2. Rubric Builder

The Rubric Builder[1] is the online version of a previously standalone software tool to create rubrics. Although it is now an online tool its primary task is still to create rubrics and finally print them out. They may be handed out to students but usually are used solely for grading by the teacher. No online evaluation or handling of grades is available. Consequently no student feedback is provided at all through the help of the system. What makes the system remarkable compared to other simple rubric creation tools like the Rubric Machine[2] is the integration of curricula. Specifically the curricula of the Ontario school system[3] are supported. Criteria are grouped into the four categories *knowledge and understanding*, *thinking*, *communication*, and *application* defined by the Ontario curricula achievement charts. Each criterion has a fixed number of four levels. The other direct reflections of the curricula are the default list of courses and lists of expectations where those are to be selected which the rubric will evaluate (see Figure 3.4). The provided lists of default criteria can be very long and cumbersome to select. The provided default level descriptions can be edited or alternatively own criteria created. The final rubric can be stored for later retrieval and editing or printed out in a grid format either sorted with levels 1-4 or 4-1 depending on the teachers preference.

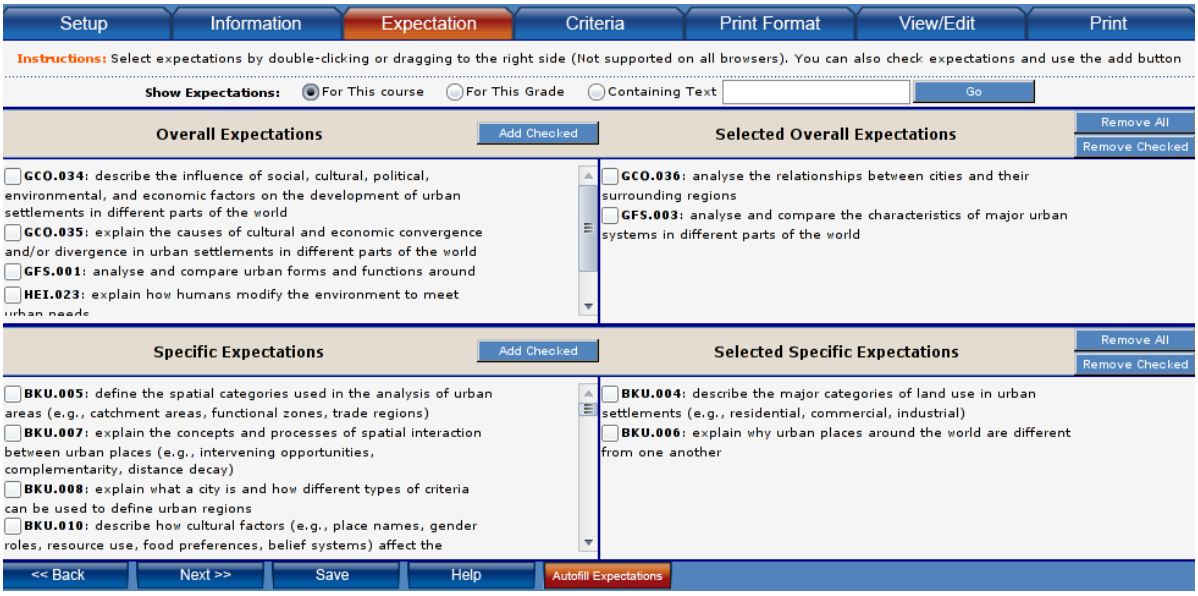


Figure 3.4.: Rubric Builder Website Screenshot, Ontario curricula expectations (Gateway Software Productions, 2006)

[1] http://www.rubricbuilder.com

[2] http://landmark-project.com/rubric_builder/index.php

[3] Curricula are available online from http://www.edu.gov.on.ca/eng/document/curricul/curricul.html

3.2. Semi-Automatic

The term semi-automatic is defined differently by authors (see also 2.3). As the main aspect in this review is automatism of assessment the following definition is used. Semi-automatic systems need to provide some kind of automatism which is not completely dependent on human interaction to assess each single essay but grades and feedback ultimately depend on the teacher. Systems automatically evaluating certain aspects of an essay presenting the results to the teacher to review, correct and extend them fit this definition. Also hybrid approaches were some aspects are evaluated fully automatically but teachers are forced to review others and manually add feedback are covered as well. Semi-automatic systems are often found in computer-science class assessment where programs or source code are automatically evaluated and then manually reviewed by teachers (Ala-Mutka & Järvinen, 2004; Jackson, 2000).

on such automatic testing to reduce the workload of teachers. The system provides management of grades and

3.2.1. Writing Roadmap 2.0

Writing Roadmap 2.0[1] by CTB/McGraw-Hill is a commercial online program for continuous essay writing practice and therefore marketed for formative learning settings. For each essay a holistic score related to a holistic rubric is calculated along with analytic scores for Ideas and Content, Organization, Voice, Word Choice, Fluency and Conventions. Students enter their essays online and depending on the configuration can use the tools hints (For example hints on how to improve the introduction), tutor (spelling and grammar, see Figure 3.5), thesaurus, and tree (Grammar tree for a selected sentence) depending on the configuration.

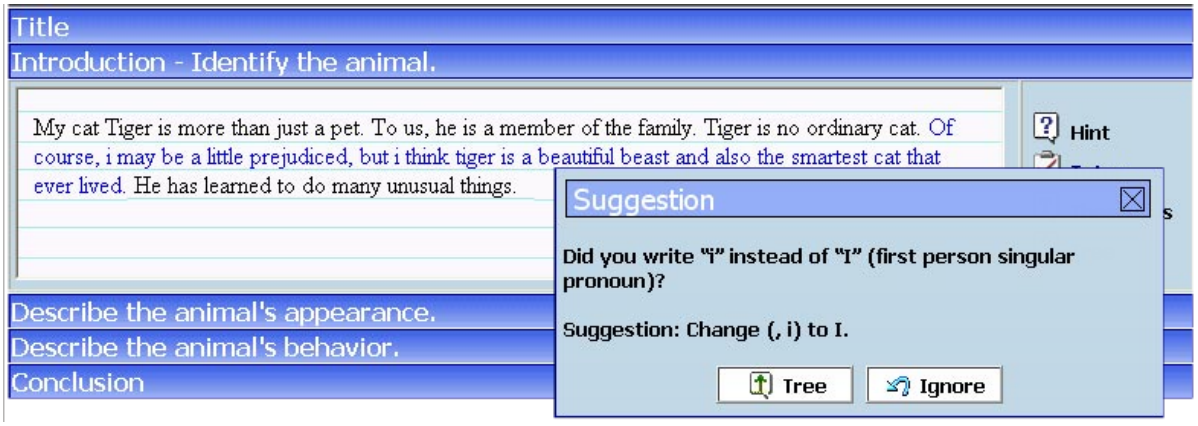


Figure 3.5.: Writing Roadmap 2.0 Student Trainer Grammar Tool, website screenshot (CTB/McGraw-Hill LLC, n.d.-a)

[1] http://www.ctb.com/ctb.com/control/productFamilyViewAction?productFamilyId=459&p=products

Essays are not entered as a continuous text but entered in sections as introduction and conclusion (see Figure 3.5). After students submitted their essay they will see the automatic ratings (see Figure 3.8). Figure 3.7 suggests that the automatic feedback is rather formulaic using teacher supplied section titles and generally explaining the analytic rating levels.

The system differentiates four different essay writing styles (Narrative, Informative/-Expository, Descriptive, Persuasive) and allows to select a overall range of achievable points from 4 to 6 for the scoring rubric. To create a new assignment teachers enter an essay prompt and select the number of sections the student has to use to organize the essay. Also the grade level of the assignment needs to be set which supposedly effects the automatic evaluation. The system can function fully automatic but contains features for teachers which lead to a semi-automatic workflow. Teachers can manually add feedback comments to students and unscore an essay allowing a student to edit and resubmit the essay. Teachers can override the holistic score the essay receives but cannot change the scores calculated for the six automatically calculated analytic scores (see Figure 3.6). It is not possible to adapt the analytic evaluation or add comments specific to the evaluated traits.

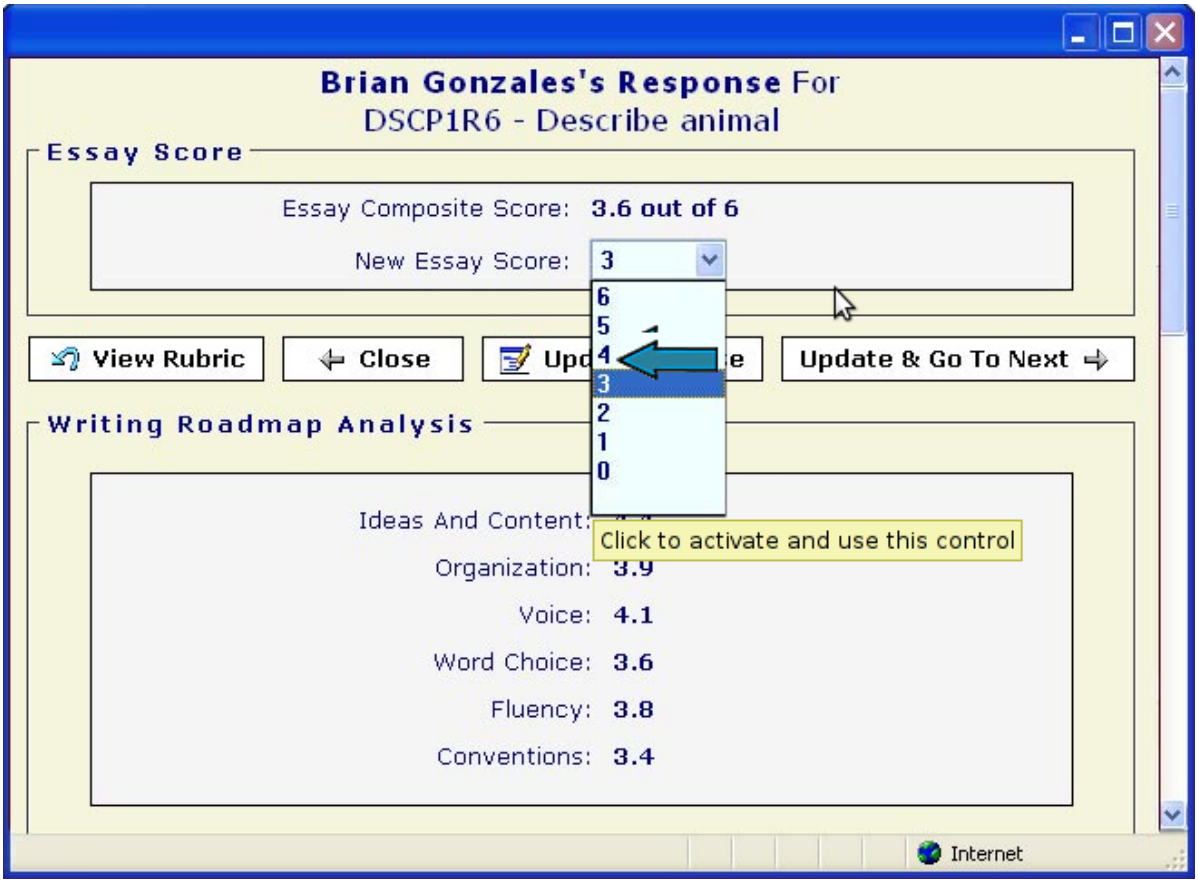


Figure 3.6.: Writing Roadmap 2.0 Teacher Trainer Scoring, website screenshot (CTB/McGraw-Hill LLC, n.d.-b)

Rich and Wang (2010) evaluated two case studies and reported the "largest score gains for low performing students" who practised at least 4 times with the system in preparation for a summative test. In the case of Chinese students studying English as a second language they found that the automatic ratings alone are "not sufficient for improving student writing".

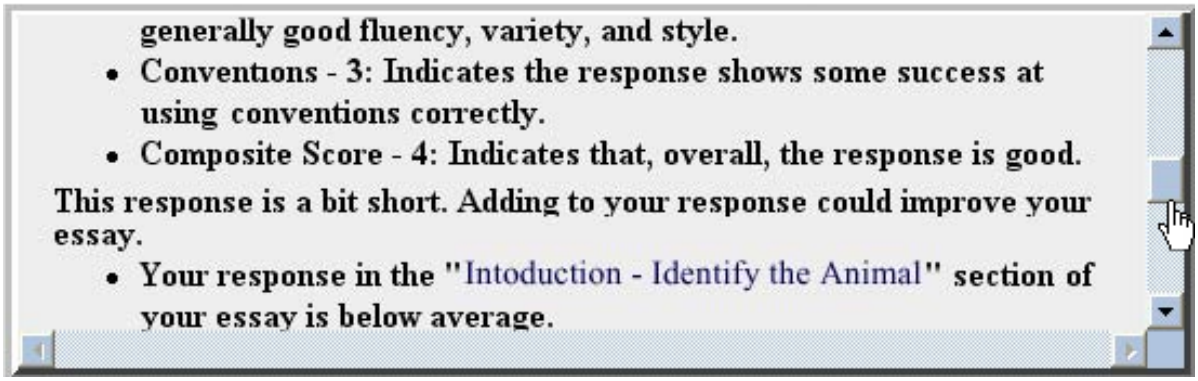


Figure 3.7.: Writing Roadmap 2.0 Student Trainer Grammar Tool, website screenshot (CTB/McGraw-Hill LLC, n.d.-a)

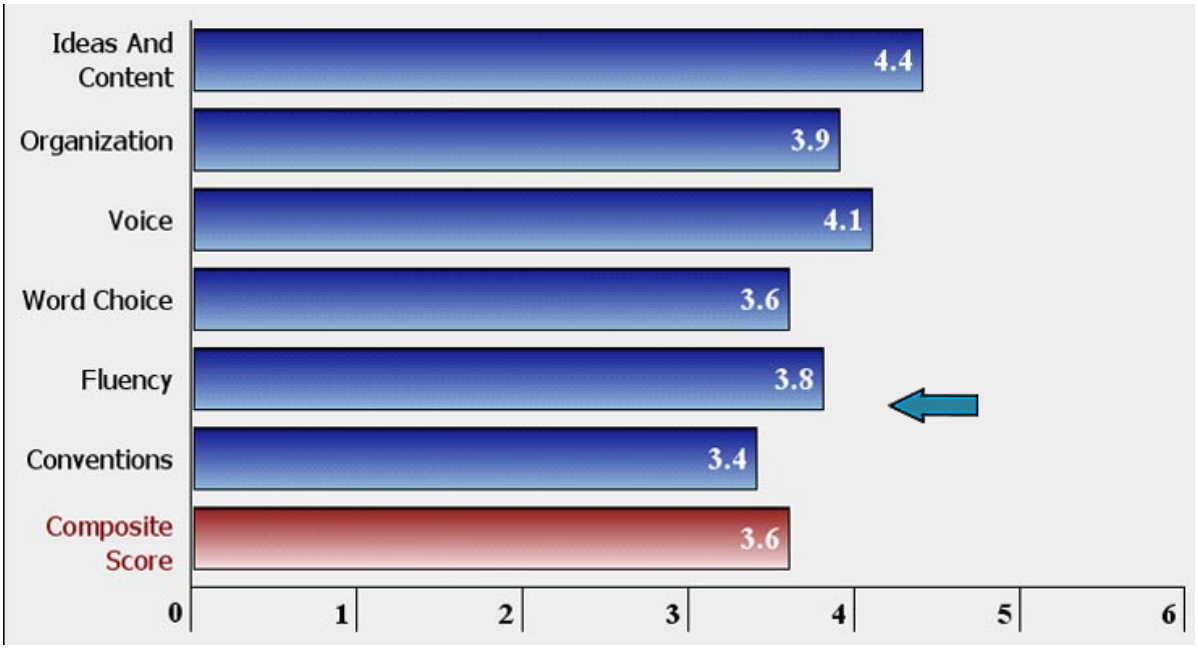


Figure 3.8.: Writing Roadmap 2.0 Student Trainer Grammar Tool, website screenshot (CTB/McGraw-Hill LLC, n.d.-a)

3.2.2. Plagiarism - Turnitin®

Detecting plagiarism is a time consuming task for teachers resulting in great benefits by assistance from automatic solutions. Several services exists which can be grouped into the two classes "local" and "global" (Butakov & Scherbinin, 2009). Local solutions search through a provided set of documents usually within a course or university without checking internet resources. As local plagiarism is common (Austin & Brown, 1999; Teferra, 2001) these solutions can detect many cases of plagiarism within a class or course over different semesters. Global solutions incorporate internet resources and may feature cross institutional databases. Depending on the academic conduct code the stakes for students in plagiarism can be very high. Some types of plagiarism like rephrasing can be more tricky to detect reliably compared to obvious direct copy and paste cases. Depending on the prompt essays delivered may have a high similarity due to a domain specific vocabulary, essay length and few source materials available to students. Therefore a manual review of automatically detected plagiarism cases has to be done in most cases hereby fitting the semi-automatic definition. As generally plagiarism solutions are specialized to this single task (Shortis & Burrows, 2009) one prominent solution is reviewed exemplary for these type of automatic essay assessment.

Turnitin® is a commercial online solution by iParadigms, LLC aiming at teachers and institutions. Additional services by the same company are offered for students (WriteCheck), publishers and researches (iThenticate®). Turnitin has been chosen not only because it is well known but as it also offers a grading solution called GradeMark as well (Shortis & Burrows, 2009).

Turnitin compares submitted papers with its own database of stored essays, journals, books and web pages. Turnitin uses proprietary algorithms generating fingerprints of the works which are then compared (Gruner & Naven, 2005; Blake Dawson Waldron, 2004). In September 2010 a newer version of Turnitin integrating GradeMark and PeerMark tools was released (*GradeMark Paperless Grading Improvements*, 2010). GradeMark utilizes a previously defined rubric for manual grading of the essays by teachers and offers possibility to add feedback to the essays. If the teachers used PeerMark for peer review by the students the results are available to the teacher while grading as well as the results from the plagiarism check. No indication for further automation aside from the plagiarism check is given in the product description except for administrative grade handling and integration into learning management systems like Blackboard, Moodle and WebCT.

As with any plagiarism checker the results delivered have to be manually reviewed for false positives by teachers. Therefore although the detection itself is done fully automatic the system is considered semi-automatic as a manual review step is needed in cases where an essay is flagged positive for plagiarism. If an essay is not flagged it still can contain undetected plagiarism which teachers could detect for example by writing style changes in a manual review. The GradeMark component is a manual rubric-based approach for grading but incorporates the plagiarism checker as an assistance tool in grading.

The practice to store the submitted students papers has raised concerns about privacy

and possible violation of student copyrights especially as this aids the company's profit (Foster, 2002). A ruling by the United States Court of Appeals for the Fourth Circuit denies copyright violations of the company mainly due to the fact that student have to accept a "Click-wrap Agreement" before submitting their work (*A.V. v. iParadigms, LLC*, 2009). Students at McGill University have successfully challenged compulsory submission to Turnitin (Churchill, 2005; CBC News, 2004) alongside other universities critical of the service (Osellame, 2006).

3.3. Fully Automatic

Fully automatic systems do not require human interaction to produce a final grade for a given student work with the exception of preparation necessary to set up and initiate the grading process. This definition includes systems which require a certain amount of assignments to be human graded in the preparation phase and then will grade the rest automatically.

Most systems on essay evaluation published in literature are intended for fully automatic evaluation. Different underlying techniques are utilized and provided feedback varies greatly. In this section the most prominent systems in respect to the published literature and practical real-life usage are reviewed.

3.3.1. Project Essay Grade™

Project Essay Grade™ (PEG)[1] is acknowledged as the first automated essay scorer with the underlying research first published in 1966 by Page. Page (2003) contended that an outsider would not have been able to differentiate the computer and human performance when presented the correlation values. The theory behind PEG assumes that *trins* related to the intrinsic variables representing the quality of the work can be measured indirectly by substitutes called *proxes* (L. Rudner & Gagne, 2001). A multiple regression equation is developed by using a large set of essays scored by humans which then can be applied to new essays to predict the score a human would give. PEG is therefore the first fully automatic essay grading system which has been published. Though not all variables used have been published by Page it can be concluded that the system mainly uses surface linguistic features and does not judge the content (Chung & O'Neil, 1997). PEG is only used to determine scores when assessing a large amount of essays and not to provide any feedback to students.

[1] A demo has been reportedly available at Indiana University Purdue University Indianapolis `http://134.68.49.185/pegdemo/` but could not be accessed by the author successfully. A short report including a screenshot can be found here `http://www.cs.nott.ac.uk/~smx/PGCHE/peg.html`

3.3.2. Intelligent Essay Assessor™

Landauer, Foltz, and Laham (1998) applied Latent Semantic Analysis (LSA) [1] to essay grading. LSA uses the context of words to determines the similarity meaning of them and represents the analysed documents in a matrix semantic space. It it then possible to calculate the similarity of a new document to previously analysed documents therefore deriving a grade for the new document if grades for the most similar previously analysed documents exist.

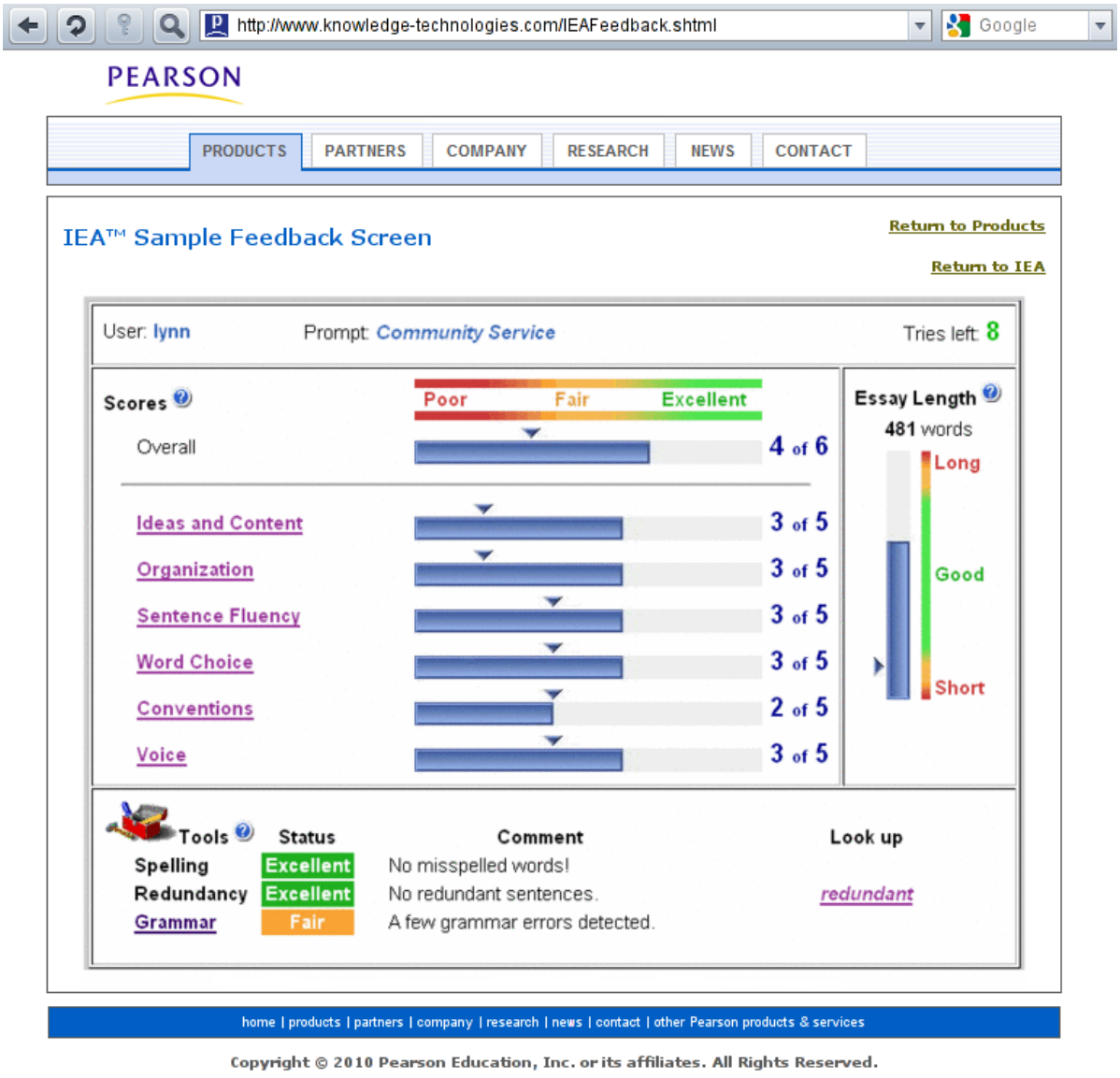


Figure 3.9.: IEA™ Demo, website screenshot
(Pearson Education, n.d.-b)

[1] http://lsa.colorado.edu

The Intelligent Essay Assessor™ (IEA™)[1] by Pearson Education is a commercial application of the LSA method. IEA™ needs about 100-200 graded essays for the training phase (Palmer et al., 2002; Pearson Education, n.d.-a). As further essays can then be graded autonomously by the system IEA™ classifies as a fully automatic system.

The online demonstration (see Figure 3.10) of IEA™ provides a list of possible spelling errors, grammar errors and a short diagnostic textual feedback on Audience & Purpose, Organization, Elaboration and Use of Language for feedback and assigns an overall score between 0-6. The textual feedback often contains "may" and therefore can be perceived as vague.

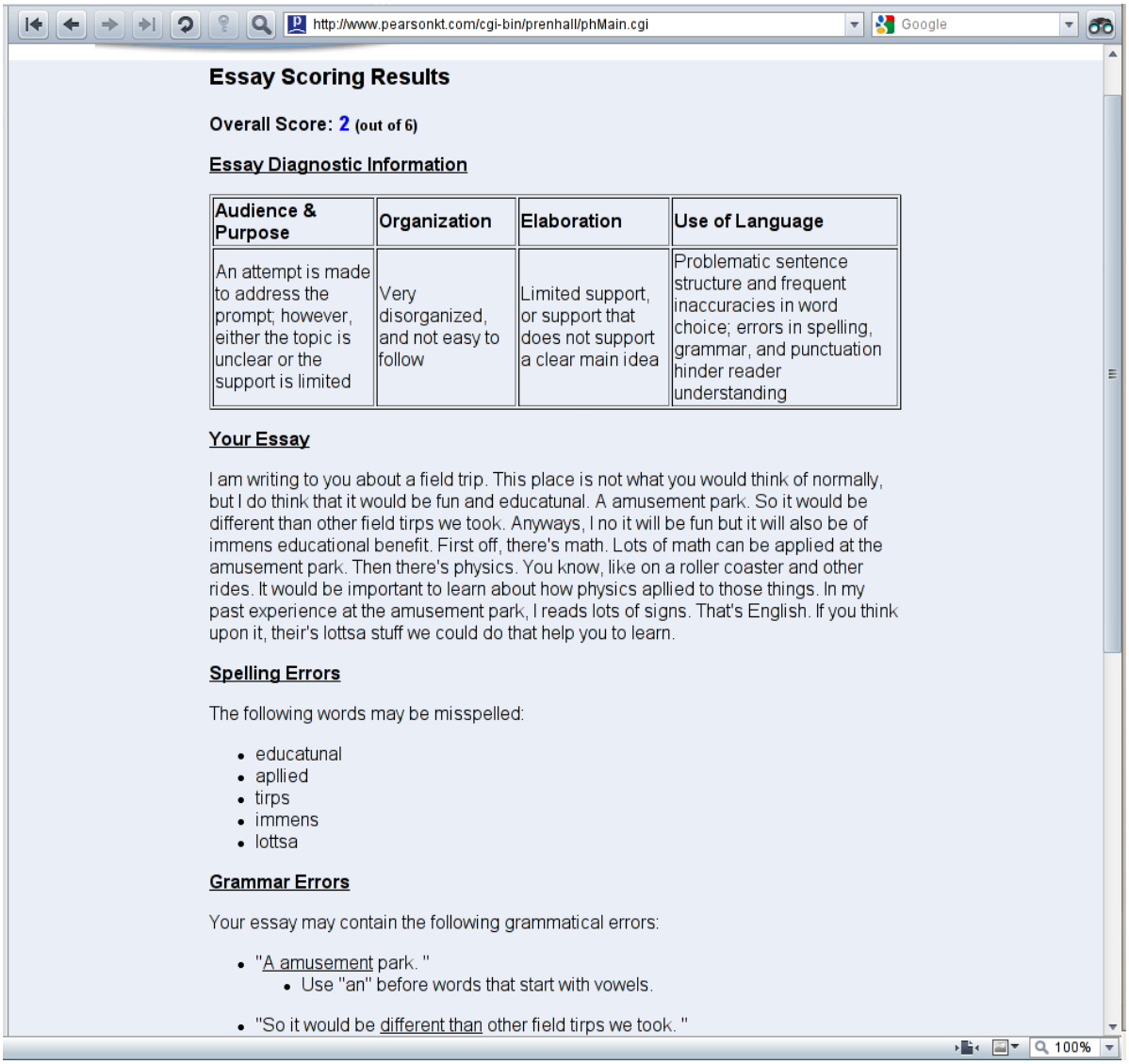


Figure 3.10.: IEA™ Demo, website screenshot
(Pearson Education, 2007)

[1] http://www.knowledge-technologies.com/prodIEA.shtml

This online demonstration seems to be a bit outdated as a sample feedback screen (see Figure 3.9 provided on the website suggests. It shows the analytic scores on the six traits ideas and content, organization, sentence fluency, word choice, conventions and voice used to grade the essay. Again it is possible to get feedback on spelling, grammar and redundancy and additionally essay length.

3.3.3. E-Rater® and Criterion

E-rater is a scoring engine developed by the Educational Testing Service (ETS)[1] using a "*hybrid feature identification method*" utilizing statistical derived variables and Natural Language Processing (NLP) including rhetorical structure, syntactic structure and topical analysis (Burstein et al., 1998, para. 1). According to Burstein (2003) it uses a corpus based approach for building the NLP model and requires graded example essays to build the actual scoring prediction model used to assess further essays. E-rater therefore classifies as an fully automatic assessment system.

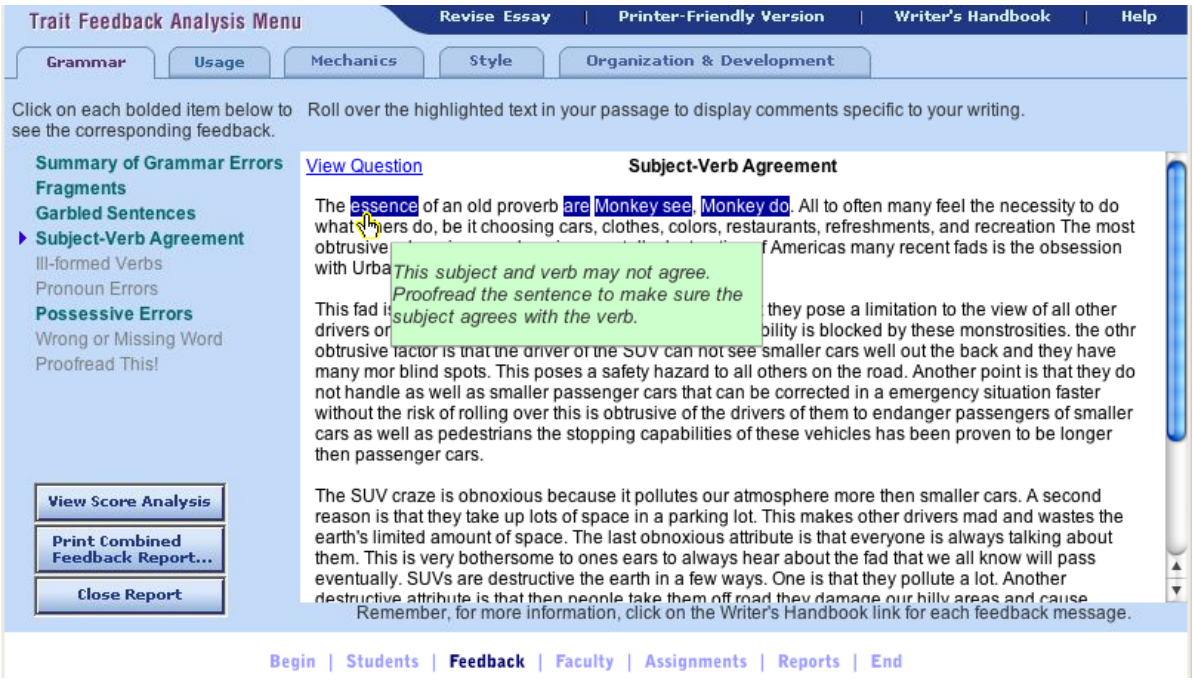


Figure 3.11.: Criterion Demo, website screenshot online flash demo (Educational Testing Service, n.d.)

The engine is used, for example, in the evaluation of the Analytical Writing Assessment in the GMAT as a second judge (Quinlan, Higgins, & Wolff, 2009; Burstein, 2003; *GMAT® Scores and Score Reports*, n.d.). Another application of E-Rater is Criterion, an online writing evaluation service intended for instructional use. It uses the e-rater engine to generate the holistic score and a separate engine to provide feedback (Burstein,

[1] `http://www.ets.org`

2003). Diagnostic feedback is provided by the Critique™ tools about grammar, usage, mechanics, style and organization & development. A student can select the feature to review and the relevant section of the text will be highlighted. In a pop-up window a short explanation or hint will be displayed (see Figure 3.11). A general writers handbook is provided which includes examples for common errors and improvement tips and the relevant section is linked to the currently reviewed error type. Lectures can add notes to an essay which a student then can read and reply to them for further clarification. Several statistics are available for lecturers to review the overall performance of a class.

3.3.4. IntelliMetric® and My Access!®

IntelliMetric® is a commercial scoring engine developed by Vantage Learning[1]. Elliot (2003) has reported that IntelliMetric® "*is based on artificial intelligence (AI), natural language processing and statistical technologies*" (p. 67). Compared to other scoring engines fewer details are known as "*Vantage's corporate strategy*" is to treat details as a "*proprietary trade secret*" (L. M. Rudner, Garcia, & Welch, 2006, p. 6). The engine groups the evaluated features into the five groups *focus and unity*, *development and elaboration*, *organization and structure*, *sentence structure* and *mechanics and conventions*. To build the scoring model the system needs around 300 previously scored essays for training (Vantage Learning, 2005). The model then will be tested against a smaller set of essays with known human scores which were not included in the training to validate the model which then can be used to determine a holistic score for novel essays. The system is applicable to other languages than English like Spanish, Hebrew and Bahasa (Elliot, 2003).

My Access!® is a web-based instructional writing tool aimed at helping students to develop their writing skills (Vantage Learning, 2005). The following description is based on a web based flash video demonstration[2] of the system as no real demonstration account is available fur public review. For lecturers statistics about the performance of a student or a whole class are available. The feedback for students is grouped into focus, development, organization, language use and mechanics & conventions (see Figure 3.12) differing slightly from the naming of the underlying engine. A message center allows lectures and students to communicate about the written essays.

[1] `http://www.vantagelearning.com/school/products/intellimetric/`

[2] `http://www.vantagelearning.com/school/demos/demos_myaccess_overview_skin`

Figure 3.12.: My Access Demo, website screenshot flash online demo (Vantage Learning, n.d.)

3.3.5. MarkIt

MarkIT is an AEG system developed at Curtin University of Technology protected by provisional patent application in Australia. The developers claimed that one model answer covering all relevant content is enough to specify the required content but that the results are not completely satisfying in comparison to other systems (Williams & Dreher, 2004). For the current system which uses a regression equation for scoring the authors recommend to use around one hundred training essays graded by at least one human rater and a model answer to build the equation during the training phase which yields satisfying results (Williams, 2006) .

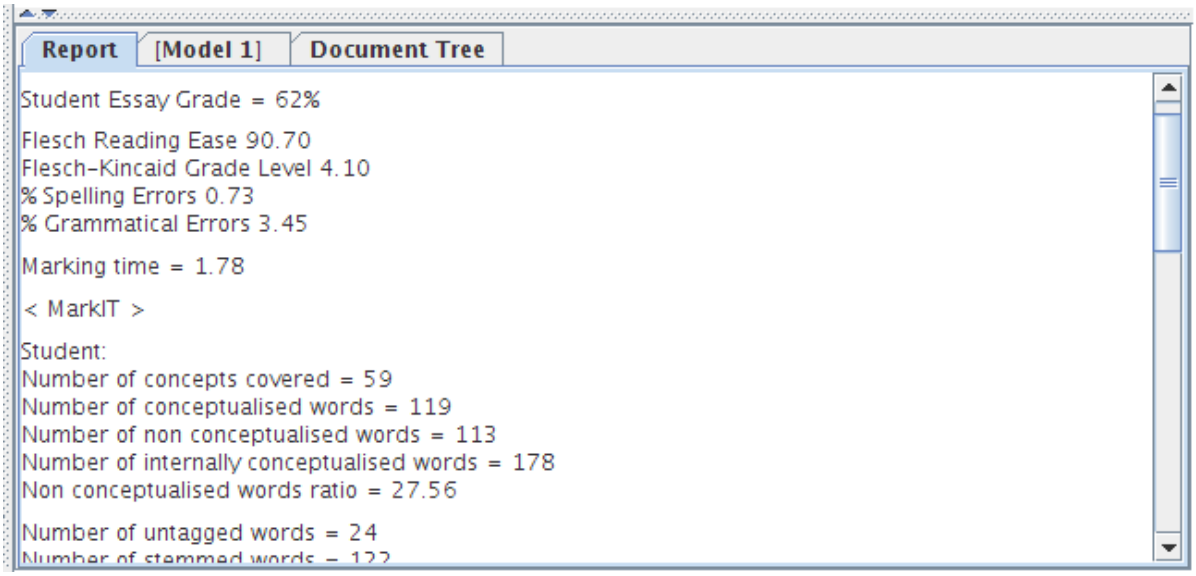


Figure 3.13.: MarkIT Demo, textual report
(Blue Wren, n.d.)

After the training phase is finished the remaining student essays are automatically graded classifying MarkIT as an fully automatic system. According to Williams the system uses chunking to determine noun phrases and verb clauses to build a representation of the semantic content. This is similar to LSA as in both approaches the essay words are used to derive the contained knowledge (Williams & Dreher, 2004). The system uses a thesaurus to reduce the concept space to 812 root concepts. More details about the knowledge representation algorithm can be found in (Williams, 2007).

An online demonstration[1] is available to evaluate the system but no upload of essays is possible[2]. The online system contains a job list comprised of one entry per evaluated assignment. To create a new job the user needs to upload human marks, models answers, training essays and student essays in separate archives. Finished jobs can be reviewed with an applet per essay and a textual report covering each essay can be downloaded. The report contains the overall grade, summaries of spelling and grammar errors and several statistics from the MarkIT algorithm used to determine the content. The report is therefore rather lengthy and a bit confusing as statistics of the algorithm are included (see Figure 3.13 for a partial sample for a single essay). The applet allows to show each essay and the model answer online. A bar graph compares the concepts covered in the model answer and the student essay. By clicking a concept in the bar the words associated with in the essay are highlighted (see Figure 3.14). Dreher (2007) has contended that this feedback can be used as formative feedback if available to the student.

[1] On the website `http://www.essaygrading.com` the actual demo is not found under the tab online demo which points only to a video but under the tab My Account when clicking on the guest account link.

[2] The upload form is provided but fails with an exception when trying to upload essays

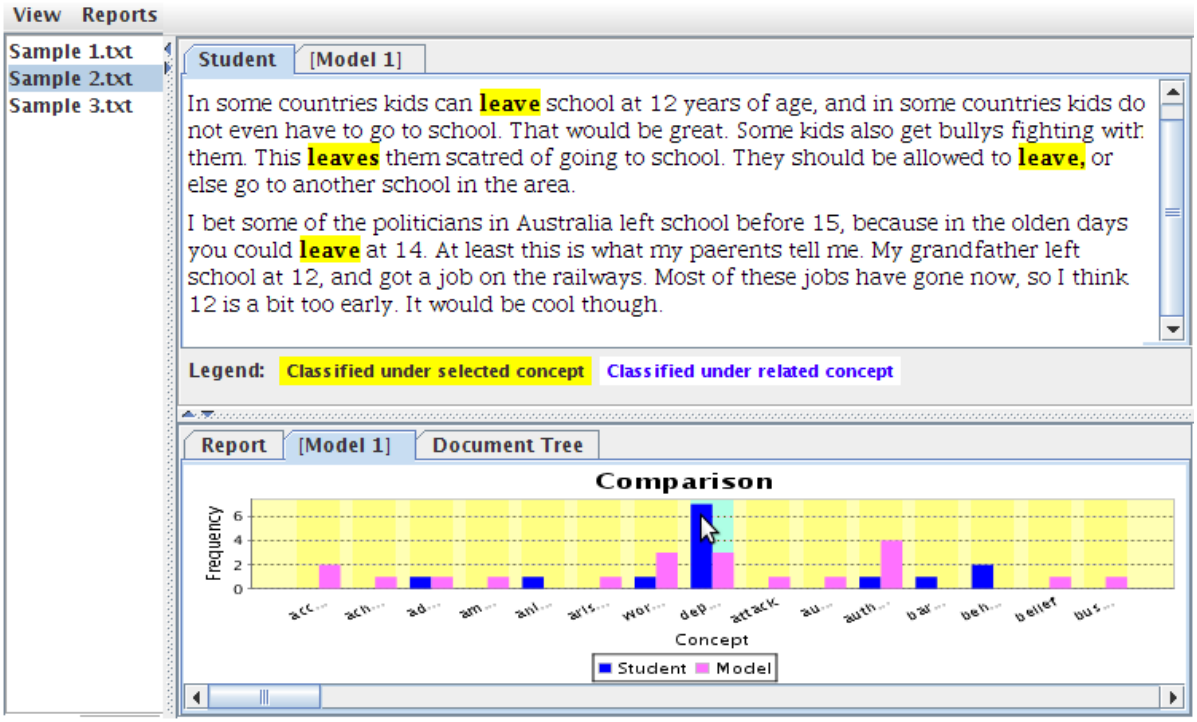


Figure 3.14.: MarkIT Demo, concept report
(Blue Wren, n.d.)

3.3.6. e-Examiner

The determining feature of Short Free Text Answers (SFTA) is actual content whereas in essay evaluation also others aspects like style may be considered important. Depending on the specific essay prompt the content in the answer essay can be very important. It is therefore of interest to look into SFTA assessment systems when considering essay grading approaches. One such SFTA evaluation system is e-Examiner which has been reported by Gütl (2008b). It uses GATE[1] for natural language preprocessing and ROUGE[2] characteristics as a hybrid approach to evaluate SFTA (Gütl, 2008b). SFTA are typically used for factual science questions. As the answer content consists of a narrow range evaluation criteria for wrong and correct answers exists resulting in objective assessment (Pulman & Sukkarieh, 2005). The main idea of the approach is to determine the similarity of a given answer to the correct reference answers using different algorithms and combining the results into a final score. Gütl (2008b) has reported correlation factors comparable to other approaches.

[1]General architecture for text engineering `http://gate.ac.uk`
[2]Recall-Oriented Understudy for Gisting Evaluation

3.4. Summary

The review revealed two opposing directions of existing systems: systems based on a rubric with marginal or no automated features and fully automatic essay grading systems, some of which incorporate features similar to a holistic rubric. In between the manual approaches and the fully automatic solutions, the semi-automatic systems generate partially automatic feedback while keeping the possibility of teachers to add further feedback manually. Semi-automatic systems either allow teachers to actually mark the assessment manually or at least permit to override the final result the automatic evaluation may deliver. The fully automatic systems can be of advantage in settings where hundreds or thousands of essays need to be scored, reducing the time needed for scoring as well as resulting in possible cost advantages. In contrast to the fully automatic systems, most of the rubric-based systems are essentially manual, therefore no real advantage is gained by teachers compared to a proper computerized approach. The feedback provided by most systems is rather limited, which makes them less usable in formative settings at the classroom level. In some cases, it seems the systems were originally developed for large scale assessment and later extended for classroom assessment. This assumption is backed by the example of Criterion (see section 3.3.3) which uses the engine developed for large scale assessment to calculate the holistic score and a second separate engine to provide feedback (Burstein, 2003). The actual feedback and grading may still relate well to each other, but an approach linking both by design in the first place would be more effective. Depending on the essay prompt, evaluation of content in the sense of concepts and ideas is a vital part of essay assessment and any approach that does not evaluate this clearly cannot be considered as reliable. In manual approaches, teachers ensure this is done, and can easily do the same in semi-automatic approaches. Performance of automatic systems in content evaluation varies greatly, or is not given at all, as in the case of the PEG system.

Many rubric-based systems are actually manual systems mimicking paper-based rubrics. Most of the reviewed systems utilize a table display for the rubrics or offer an additional list view. Systems without a tabular view are often less usable for analytical rubrics. Teachers are usually supported through rubric collections to derive new rubrics from given examples or previously self created rubrics. The Rubric Builder system was outstanding in comparison to the rest as it incorporated official curricula by displaying them and offering criteria grouping as requested in the referenced curricula. Manual systems provide no further assistance to teachers aside from basic support tools in designing an assessment and possible integrations into Learning Management System (LMS) system, removing the need to manually transfer marks into electronic gradebooks.

The fully automatic systems reviewed (PEG,IEA™,E-Rater® and IntelliMetric®) release teachers from protracted marking tasks but are limited mainly to large-scale assessment, as they require a fair amount of pre-scored essays. Validity and acceptance are still an issue especially in high-stake testing as is demonstrated in the example of the GMAT (refer to 2.3 for details). The different scoring techniques follow a common approach using approximations or correlations to represent the particular variables that would be considered by a human. These were named *trins* referring to the intrinsic vari-

ables which are measured and represented by substitutes called *proxes* (approximation) by Page and Petersen (L. Rudner & Gagne, 2001; Attali & Burstein, 2006). According to Attali and Burstein (2006), most AEG systems use a modelling procedure (linear regression, statistical models) to build an internal model to score preciously unscored essays by analysing a training set of essays with human scores to find the proxes which best predict the human scores. It is therefore not transparent to the user which proxes are actually used internally. A major problem of the fully automated system is the preparation of the required training data. For example the MarkIT system uses a model answer and a larger set of training data consisting of human graded submissions in the range of one hundred essays to configure the algorithm (Williams, 2006). In the study by Shermis, Mzumara, Olson, and Harrington (2001) 1293 essays where scored by 6 human raters to build the statistical model used by PEG. Providing this training material requires a great effort and therefore limits the use of AEG systems to classes with a high number of students.

Semi-automatic approaches are often found in plagiarism detection solutions. They support teachers in detecting possible plagiarism but do not replace teachers. In many cases flagged plagiarism needs to be reviewed carefully by teachers to reliably detect student misconduct. The solutions cannot guarantee to detect all possible forms of plagiarism and therefore teachers' reviews are still necessary in high-stake situations. Besides plagiarism checkers, more general solutions (as Writing Roadmap 2.0) focusing on complete essay assessment are available. They generally evaluate a fixed set of criteria providing a holistic score and sometimes analytical scores for the fixed criteria set. These systems may offer automatic evaluation but leave the final decision to the teacher by allowing a manual override of the result.

The system review clearly indicates a lack of solutions applicable on the classroom level which offer clear advantages over manual approaches. The existing solutions either do not support teachers well or are limited, rigid automatic solutions. AEG systems are inflexible, as it is not possible to define a chosen set of criteria against which the essay will be evaluated. No system intended solely for essay scoring utilizes an analytical rubric to communicate the assignment requirements and evaluation details between teachers and students.

Concluding from these findings, the potential is given to develop a rubric-based solution combing aspects of existing solutions and at the same time offer more valuable feedback to teachers and students. This solution needs be more flexible than current systems so that criteria can be specified by teachers. To maximize the benefits of the new system it should be applicable at the classroom level and possibly in formative settings as well, as this is supported only by very few systems. As the number of students is much lower at the classroom level, a semi-automatic approach seems feasible to reduce the workload of teachers and to provide timely results while ensuring validity as teachers make the final decision about the actual score.

4. Design Approach

The preceding chapters provided the theoretical background for a new type of essay grading system, which is rubric-based. The literature review has revealed the general importance of assessment in the learning process and even more in formative education environments. Rubrics were introduced as an appropriate scoring tool for essay assessment, both in summative and formative settings. Formative assessment is time intensive for teachers and therefore software that supports teachers in the process can be of great help to establish formative practises within a restricted-time budget. Existing solutions were found to either be inflexible, to not support teachers well enough or in the case of many AEG systems to be limited by application only to large class sizes.

Hence the conception of this project is to develop a system which can be applied at the classroom level, both in summative and formative educational settings. The grading tool utilized is a rubric, which can be flexibly designed for each assignment task. The system supports teachers by providing automatic analysis and possible automatic ratings for the selected rubric criteria to speed up the evaluation process. In the next section, key issues for such a system are discussed.

4.1. Identified Key Issues

Correlation between calculated grades and grades given by a human assessor is not the only critical criterion for automated evaluation features found in semi-automatic and fully automatic systems. Kakkonen and Sutinen (2008) have proposed a comprehensive evaluation framework consisting of a set of criteria. The framework is applicable to semi-automatic assessment systems for free-text responses. One key aspect of the framework is that not only preciseness but also provided feedback and human effort are evaluated for each criterion where applicable. Deriving from these criteria, one can conclude that the following aspects of an essay grading system with some automatic features are most important:

- validity
- low human effort
- feedback (analytical)
- traceability (understandable results)

These key issues are reviewed in the following sections. Additionally, the issue of class size limitation as seen in current AEG systems is discussed.

4.1.1. Class size

To apply an evaluation system at the classroom level with student numbers below one hundred, it is necessary that only a small training set or ideally one model answer is needed to configure the essay grading system. This is especially vital when the system should be applicable in formative assessment. As this type of assessment occurs more frequently any effort needed to prepare the system multiplies the human effort or prohibits effective usage of the system in formative settings.

4.1.2. Human effort

A major goal of essay grading systems is to save time and allow teachers to focus on other tasks. As previously outlined, extensive effort in providing training material to the system must be avoided as this effort is irrespective of the actual usage in teaching whether it is summative or formative. Generalizing this fact, it is a vital goal for such systems to minimize time spent on tasks aside from the actual grading. One example is the student submissions. A system which forces manual importing or even the conversion of student submission before they can be graded wastes time. Therefore the effort for teachers as well as students should be minimized wherever possible, not only focusing on the preparation of training material. In conclusion, this requires students to submit the essays electronically.

4.1.3. Feedback

One highly weighted requirement of an assessment system is the feedback provided to its users. The two main user groups are teachers and students. The particular interest of the students is to understand why they received a certain grade and how they may improve their results. For the teachers, results of a whole class, not only a single student, are important to adapt instruction.

One can argue that a system used for formative assessment needs to provide richer feedback to the users than rather than one used only for summative assessment. For students and teachers the progress between assignments is of interest. Progress can only be measured reliably if the same or a comparable scale is used for each assessment.

While general recommendations in the form of: "You have too many spelling errors. Make a habit of proofreading your text before submitting it" are easy, it is harder to automatise a more detailed feedback on other criteria. Chen and Cheng (2008) conducted an study with My Access!® using the IntelliMetric engine. Regarding the provided feedback, the study reported that "*around 50% of the respondents in each class considered it to be of no help*" as it was conceived as "vague, abstract, unspecific, formulaic and repetitive" (p. 104).

An analytical system offers chances for the issues of progress detection and vague feedback. In an analytical analysis several independent scales are used. Therefore students and teachers can at least measure progress in certain aspects directly, even if the overall assessment is slightly different for each assignment, as long as some of the scales are

used at every assignment evaluation. Feedback summarizing every aspect into a single rating as it is done in holistic scoring has a tendency to be more unspecific as there is no formal differentiation between different criteria. Analytical assessment inherently forces feedback to be more specific as separate feedback for each criterion needs to be given.

4.1.4. Traceability

To fully understand the provided feedback, one needs to understand how the work is assessed. As previously outlined, automatic systems typically use training data to internally configure the evaluation algorithms. This configuration is a task usually only solvable for the involved researchers and therefore generally not exposed directly to the end user. The hidden internals are expressed in construct objections as contended by Chung and O'Neil (1997). Resulting from that, teachers as well as students conceive the system as a black box design as they will not be familiar with the current state of research behind the implemented approach. To foster their acceptance, teachers need assurance of the validity of the system as well as the feeling to be able to control it to a certain extent, thus avoiding construct objections. Again an analytical system addresses this problem at least partially, as it is more clear which criteria are evaluated.

4.1.5. Validity

Validity is extremely important for the use of automatic evaluation features in assessment systems, as is shown in many studies (Gütl, 2008b; Ben-Simon & Bennett, 2007; Williams & Dreher, 2004; R. James et al., 2002). This applies not only to fully automatic systems but also to the automatic features incorporated into semi-automatic systems to be useful to teachers. Chapter 2.3 contains a more detailed discussion on validity of automatic systems. Reliability and absence of bias are two other important aspects of assessment (Lang & Wilkerson, 2008) related to overall system soundness. A properly designed automatic system will be reliable and only changes in used algorithms would lead to differing results over time. Similarly, absence of bias can be assumed, as the system will not have any personal affections to students or groups as teachers might have.

4.2. Conception for an Essay Grading System

Compared to existing systems a new essay grading approach has to provide more feedback and should be applicable to small class sizes to deliver notable advantages. Rubrics provide inherent feedback and are a good scoring guideline for teachers to timely assess student assignments in a consistent way. They are therefore a suitable tool for small class sizes and the system should be centred around a rubric-based approach. This idea is also found in the proposal for a flexible e-Assessment System by AL-Smad and Gütl (2008). The authors propose to use rubrics to "*assess learners' performance against a scoring scale and specific criteria*" allowing "*the educators to design their own rubrics*".

The authors describe a more generally applicable system, not specific to certain assignments such as essays. In this project the idea is applied in the context of essay grading. The idea is backed by a study by Anglin, Anglin, Schumann, and Kaliski (2008). The study explored the application of electronic rubrics compared to paper based solutions. Significant efficiency improvements have been reported while preserving student satisfaction regarding the received feedback. The authors have concluded that an integration of electronic rubrics into learning management systems is recommended.

A study by Andrade and Du (2005) has reported that students actively use rubrics to gain higher scores by evaluating what is expected, revising their work and reflecting on the feedback provided through the graded rubric to improve future work. This is supported by a meta study by Jonsson and Svingby (2007), which has contended that learning, self-assessment, instruction and teacher feedback are facilitated through rubrics since they contain expectations and criteria explicitly. The study has suggested that analytical topic-specific rubrics are the best choice.

The proposed system is therefore based on an analytic rubric. It supports teachers by providing automatic analysis and possible automatic ratings for the selected rubric criteria to speed up the evaluation process. The final grade decision is left to the teacher resulting in a semi-automatic solution applicable at the classroom level. High-level requirements for such a rubric-based essay grading system are discussed in the next section followed by a further outline of the system.

4.3. High-level Requirements

A rubric-based essay grading system is a flexible system allowing teachers to design rubrics specific to the assignments given by selecting and defining the criteria which will be evaluated. To reduce the workload, automatic features assisting teachers in grading are provided. Results as well as the rubrics themselves can be published. Based on this idea and the previously outlined key issues, the following requirements for a new rubric-based essay grading system are derived. They are grouped into general and rubric specific requirements and every requirement is briefly specified in the following sections.

Rubric specific Requirements

Using a rubric to address the named key issues results in certain requirements related to the construction and usage of the rubric. The major requirements are briefly specified below.

Analytic Analysis Analysis should be done analytically providing feedback for each observed aspect separately. Each criterion level must be specified in an understandable way.

Assignment Specific Definition Rubrics are defined specific for each assignment. Each rubric can be further defined by providing positive and negative example solutions.

Multiple Criteria (Extensibility) Multiple criteria must be available for the analytical rubric. Therefore the system should be easily expandable with new analysis features. The architectural design should cover the case where new variations of algorithms are implemented without compromising existing results.

Target Educational Level The system should generally be usable at any educational level up to and including higher education. To achieve this, different analysis algorithms may be used at distinct educational levels.

Automatic Evaluation Rubric criteria should be automatically evaluated with the possibility of a manual override by the teacher. Criteria which cannot calculate an automatic result should assist teachers, for example by highlighting relevant text parts on request.

General Requirements

More general requirements, in addition to the rubric specific, concerning the overall system such as the Graphical User Interface (GUI) considerations are given below.

Feedback Aside from the feedback for each analytically assessed criterion, comments and on-screen annotations for further feedback by instructors should be supported.

Human effort Effort both in preparing an assessment, as well as human interaction time in the actual assessment process should be minimized.

Traceability Transparency of how the system works should be given to teachers and students as much as possible considering the trade off between understandability and complexity.

GUI Ideally, the GUI should be usable with minimal or no training required.

Workflow It should be possible to resume grading easily after a long break. Clear indication of assessment status for each essay must be given. Grades must not be changeable after they have been published. It should be possible to process automatic assessment parts as background jobs to allow the user to continue work.

Integration It should be possible to later integrate the system into existing learning platforms. At the bare minimum, interfaces to import assignments, student lists and export grades need to be provided.

Privacy The system must ensure user privacy especially for students. Minimal data about students should be stored to comply more easily with legal regulations such as *Family Educational Rights and Privacy Act (FERPA)* (2004) in the US.

Formative assessment The system should be applicable in formative assessment settings. For example, support to review student progress over several assignments within a course should be provided.

Languages Both essay evaluation and the GUI should support different languages. English must be supported as the default option.

Multiple users The software must support multiple different users, therefore, valid login credentials are necessary. The system must be usable by different users at the same time (e.g. students accessing results and teachers grading). Multiple raters for one assignment should be supported for larger class sizes to split workload or to check rater performance. Supported user groups are teachers (raters), students and administration staff. Teachers are considered the main user group with full system interaction while students can only review results and administration staff can gain access to relevant data such as final marks.

4.4. Proposal for a Rubric-based Essay Grading System

A configurable rubric is used to analytically assess student essays by teachers. Teachers select and configure criteria to be evaluated for an assignment. Criteria either can provide an automatic rating or assist teachers in grading, by providing analysis of the essays. If procurable, the system should be able to function fully automatically if all selected criteria are capable of automatic evaluation. The default workflow is assumed to be a semi-automatic assessment where the automatic ratings assist teachers in grading the essays. In this workflow teachers ultimately decide about the final grade for every submitted essay. The default semi-automatic workflow for teachers consists of the following steps after the assignment has been specified outside the system:

1. The rubric is defined by selecting criteria for the evaluation. One or more example solutions are added to the rubric and the criteria are configured for evaluation.

2. The rubric is applied to the example solutions to check if automatic results are as expected. [Optional step]

3. After the submission deadline automatic evaluation is triggered if corresponding criteria providing automatic evaluation are used in the rubric.

4. Teachers review each essay, providing further feedback to students through annotations, utilizing provided analysis data by the criteria and possibly correcting

automatic ratings. Automatic ratings are corrected in two ways: either by overriding the result in the rubric, which is possible for every criterion, or by providing some retraining input by the user specific for a criterion.

5. Another automatic evaluation run is done to partly re-evaluate essays where automatic results have not been overridden by the teacher. This step only applies if some criteria received retraining input through the user in step 4. (Conditional step)

6. When all essays are reviewed, the student results can be published.

The system should offer as much flexibility as possible for designing a rubric and how it is evaluated to match different teacher needs. Criteria can be grouped (or categorized) to enhance the visual representation. As the proposal is based on a analytical rubric, a grid representation of the rubric is chosen. The system should allow teachers to select all possible methods to calculate a final score as described in section 2.5. It is therefore possible to specify the weight for each level. See Figure 4.1 for a schematic of the rubric layout and the possibilities.

The usage of a rubric directly addresses some of the previously outlined key issues. A rubric provides a high level of consistency between formative and summative assessment, as the same basis (the criteria) can be used for both tasks. A rubric specifies how the assignment will be evaluated by defining the criteria and levels, therefore making the assessment process more transparent to students and teachers. It may as well serve as a further declaration of the assignment specification. A rubric also provides feedback to students on how to improve their work through its descriptive nature (Moskal, 2000). Feedback is richer if an analytical rubric is used, as students can better understand which features a quality work contains and teachers can see which areas to target in their instruction to improve students' progress (Arter & McTighe, 2001, p. 22).

4.4.1. Criteria

A single criterion at rubric level can internally evaluate several features at once depending on the needs of the implemented algorithm[1]. In such a case one criterion at rubric level can be seen as a holistic rating comprising of these features. The first high-level feedback given by the system is which achievement level was reached for each criteria. Therefore it is describable to have criteria which evaluate only one or a small set of highly correlated features to ensure the overall assessment is analytical. The other extreme is an overwhelming number of available criteria which could confuse the teachers when defining the rubric or the student when interpreting the results. Therefore if a newer version of an algorithm delivering better results is available, only the current one should be shown when constructing a new rubric. For previously defined rubrics the older versions have to be preserved to ensure it is clear how already published grade results have been achieved.

[1]To distinguish this case *criteria* and *criterion* refer to the criteria at rubric level while the actual criteria the algorithms evaluates will be called *features*

See Section 2.5 for rubric criteria in general and a listing of common criteria in essay rubrics.

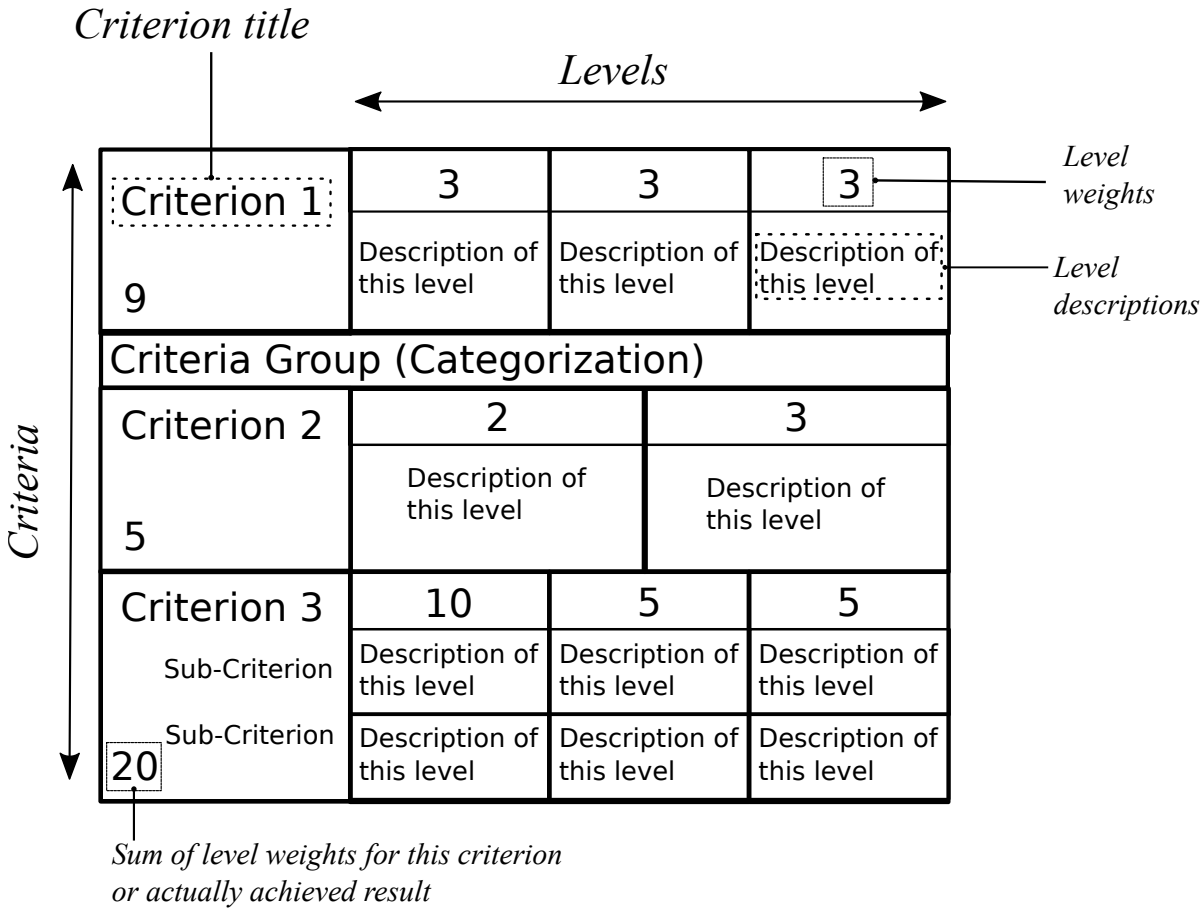


Figure 4.1.: Proposed Rubric Design

Sub-Criteria

Sub-criteria are introduced solely for the purpose of richer user feedback. They address the problem of some algorithms requiring to evaluate a small set of highly related features at once, which should be reflected in the rubric. This can be achieved by showing them as sub-criteria of the higher order criterion (for a schematic visualisation see Figure 4.1, Criterion 3). This is only useful if the evaluation algorithm can provide distinct ratings for each sub-criterion.

Plagiarism

One special criterion is plagiarism which cannot be easily represented as a regular rubric criterion. The general conduct is that plagiarism yields a zero score and possible further measures depending on the school policy. If it was only represented as a regular criterion it would influence the final score but not necessarily yield a zero score. To address this

problem, a separate plagiarism probability is calculated and evaluated separately to the general overall score. If a certain limit is reached, the overall score will be zero and the essay is flagged to be reviewed by the teachers for plagiarism. Evaluation algorithms for essays may deliver a plagiarism probability as a side effect. Therefore, each criterion can contribute to the overall probability for plagiarism. Additionally, the system allows to include special algorithms only providing plagiarism detection.

4.4.2. Validity

As outlined in Section 2.3, it is common to determine the correlation between human scores and the automatically calculated scores to validate a system providing automatic ratings. The proposed rubric-based essay grading system is a challenge to validate, since the overall result will differ depending on the actually used criteria in the rubric. This dynamic composition of criteria makes it difficult to ensure general validity. A necessary prerequisite for the overall validity is to analyse the validity for each rubric criterion. Following that logic, overall validity can be assumed if the evaluation of each rubric criterion is fully independent from each other.

In the proposed default semi-automatic workflow, validity can be assumed as teachers decide about the final ratings therefore either overriding automatic ratings or implicitly approving them. To reduce the workload, provided automatic ratings must be as concise as possible. When teachers do part of the assessment and final grading on screen, the question of validity and reliability compared to traditional paper-based marking arises. A meta-study by Johnson and Greatorex (2008) contended that the mode of presentation of texts to assess could lead to *"qualitatively different judgements"*. Contrasting a practical study comparing on-paper and on-screen rating in short textual answers by Shaw (2008) reported a high inter-rater reliability. This is backed by a later practical study by Johnson, Nádas, Bell, and Green (2009) reporting that the mode of assessment has no *"systematic influence on marking reliability"*. Although research in this area is still ongoing, there are indications that on-screen marking can yield comparable results to on-paper assessment.

4.4.3. Formative Learning Support

A valuable side-effect of the analytic design is to gather information about the progress of students. Some assessment and learning solutions track students' progress and adapt the learning path automatically (Nussbaumer, Gütl, & Neuper, 2010). While the proposed system cannot adapt the learning path directly, it can at least provide data supporting teachers in adapting teaching and assessment design. Through the usage of distinct criteria it is possible to track the progress of either single students or a whole class over several assignments for a single criterion if it is reused in different rubrics. For example if the spelling criterion is used for each assessment, teachers can check for students' progress and adapt their teaching plan if more training for the students is required in this area. Students themselves can gain insight in which areas their performance did

not improve, if they receive the analysis results and are enabled to revise their study strategy.

4.5. Summary

The proposed essay grading system is centred around a flexible rubric addressing the key issues of class size, human effort, feedback, traceability and validity. Analytical assessment increases the available feedback to students and makes the grading more transparent to both teachers and students as the evaluated criteria are explicitly stated. As the teachers design the rubric by selecting criteria, construct objections will be greatly reduced, raising the acceptance of the system among teachers. Dividing the analysis into distinct criteria makes it possible to validate automatic criteria features separately easing the validation process. Finally, the semi-automatic workflow addresses validity as teachers make the decision about the students' grades.

The system is designed to be used at the classroom level and can be used in summative as well as formative settings due to the faster assessment and provided analysis of student progress. The default semi-automatic workflow assists teachers to timely assess the essays as automatic analysis provides suggestions allowing a greater number of assignments to be handled by a teacher in the same amount of time.

Besides the actual grading, human effort is involved in preparing the assessment and is therefore defined by the complexity of configuring the selected criteria. The grading effort can be greatly reduced by the automatic ratings if they are concise.

Additionally to the analysis for grading and giving feedback to students, teachers are supported in the important task of plagiarism detection. The system design allows to integrate plagiarism detection as a specialised criterion or as a side effect of essay analysis by regular criteria. The system highlights essays that may show indications of plagiarism to teachers.

The flexible design of the system allows to extend the system with further criteria and adapt them to different educational levels. Theoretically, the system could be applied to large class sizes as well, if all used criteria provide automatic ratings. In this setting, the possibility for feedback through teachers is lost and only the analytical ratings and possible further automatic feedback are left to the students. This is not considered the default application of the system but is possible with some adaptation.

This chapter listed high-level requirements for the proposed system which will be addressed in the following chapter presenting the system design in greater detail.

5. A Rubric-based Semi-Automatic Essay Assessment Framework

This Chapter first discusses relevant additional considerations to the proposed system, followed by a reduced specification including non-functional requirements for a first limited prototype. The Chapter continues with an architectural overview of the prototype system, followed by implementation details especially covering the automatic criteria evaluation algorithms.

5.1. Additional Considerations

In extension and addition to the high-level requirements given before (see Section 4.3), supplemental considerations, like for example GUI issues and the improvement of the system, are discussed in this Section. Usability is an important issue for the acceptance and effectiveness of a software solution. This is even more true when an established traditional approach like paper-based essay grading should be replaced by a computerized solution. Humans have the tendency to stay in their trained habits, so a new approach needs not only to deliver measurable advantages but should also be as convenient as possible for the user. This is especially important in the case of a semi-automatic system where teachers are reviewing every essay.

5.1.1. GUI

Schneiderman has stated that "*usability is about understanding, stating, and serving user needs*" (Shneiderman, Grinstein, Kobsa, Plaisant, & Stasko, 2003). The user only interacts with the GUI and does not see the underlying system design. The underlying system design delivers the advantages the solution offers, but only the GUI makes it usable for the average users. In the context of essay grading one of the user's needs is to read the submitted students' essays. Traditionally essays used to be submitted on paper and teachers are still used to read and grade on this medium. When switching the medium, and therefore also the presentation of texts, the question if marking on different media could possibly lead to different results is raised. Indications that this is not an issue for assessment validity exist (as discussed in Section 4.4.2) if basic requirements are met. Johnson and Greatorex (2008) reported that accessing (navigation) and on-screen readability are significant for the performance of graders on screen. Therefore readability and navigation must be seen as key requirements for the presentation of essays. To achieve this, texts might be transformed from the original submission form.

As in essay grading also the form itself can be a criterion in assessing essays, the original formatting should be available as a second option for review by the teachers.

Annotations by teachers are extremely valuable written feedback for students and can be provided by teachers, in the case of semi-automatic grading, similar to manual approaches. Shaw (2008) reported the importance of annotations in active essay reading and that differences between on-paper annotations and on-screen annotations have been reported in some studies. Schilit, Golovchinsky, and Price (1998) contended that reading combined with critical thinking, named as *"active reading"*, is supported by the habit of annotating a text and may be impaired by restrictions of on-screen annotations compared to free-form ink annotation. Therefore, especially in the case of a semi-automatic grading system, teachers should have a wide range of annotation tools provided in the GUI supporting their individual habits.

5.1.2. Data gathering for algorithm improvement

The underlying algorithms are critical for the performance of automatic assessment. Depending on the feature to evaluate they can be hard to design and tricky to implement. To improve such an algorithm real performance data is often needed. In a semi-automatic system the manual overrides of automatic ratings through teachers are valuable data for algorithm improvement and should therefore be always recorded internally. This includes the case when an automatic rating is approved by the teacher. Due to privacy concerns only anonymous statistics about the number of overrides and algorithm configuration options may be transferred to the algorithm developers in a real usage scenario. Although the actual essays will be missing, this data can still indicate the need for particular improvements and potential research directions. In a research setting the collected data including essays can be made available to the developers providing a rich data basis for developing improvements.

5.1.3. Common basic processing

Certain text processing tasks like tokenizing, stemming and Part of Speech (POS) tagging are common operations in many information retrieval and NLP systems and therefore should be provided by the base systems. Evaluation algorithms can reuse this shared data resulting in overall system performance gains. As the underlying raw text is still available, it is possible for algorithm developers to use different routines if necessary.

5.2. Specification of a reduced demonstration system

For a first prototype of the rubric-based approach not all of the high-level requirements are necessary to evaluate the overall approach. To reduce the development time only teachers will be addressed as users. All data and most of the visualisation for students will be available and only the restricted views build out of these are left apart and can easily be added later. This similarly applies for administrative tasks like course

and user management, which can be added later or be provided through an integration into a LMS. This integration is also not necessary for a prototype and must only be addressed in internal system design decisions. Not all options like the handling of late submissions need to be implemented for a first test of the system. Multiple users will be supported for different tasks at the same time. Support for workload distribution will not be included in the first prototype but underlying support for multiple raters will be provided. Shortly summarized the following high-level requirements of Section 4.3 will not be or only partially addressed in the first prototype:

Multiple users Only teachers as users. System will allow different users to work on different tasks at the same time. Only basic support for multiple raters (e.g. no workload distribution).

Feedback Extended on-screen annotations tools can be added later as long as individual feedback can be given by teachers.

Integration No actual system integration into existing learning platforms but considered in internal design decisions.

Privacy More data may be gathered in research trials than in a real system. No artefacts of extended data gathering must remain in a release version. Ensure they can easily be removed.

Languages Prototype system will only support English.

5.2.1. Non-functional Requirements

Aside from the functional requirements more general decisions regarding the system need to be made. Therefore further non-functional requirements as platform, basic system requirements and other constraints not covered by the high-level requirements of Section 4.3 are specified now:

- The Java platform is used because of the operating system independence which includes a graphical toolkit for the GUI.

- The prototype is planned as a standalone Java application but the design should allow easy refactoring to a service oriented approach supporting different visualisation clients including webclients.

- Therefore the GUI will be developed with SWING[1] which can be transformed into an applet running inside a browser.

[1] part of the Java platform

- Internally essays should be stored as Extensible Markup Language (XML) which can be easily transformed into HyperText Markup Language (HTML) for a possible native webclient (HTML, CSS, Javascript).

- Because of the possibility of an applet version small footprint libraries are preferred if they deliver the required functionality. Libraries should ideally be available under a license which allows to distribute them in binary form with the application regardless of the final license of the system. For example the GNU Lesser General Public License (LGPL) and BSD licenses[1] are fulfilling these requirements and additionally allow linking with proprietary products.

5.2.2. Evaluation algorithms

For a real demonstration of the rubric-based approach several criteria need to be available to design a rubric. Based on the common list of rubric criteria in Section 2.5 the criteria (i) Spelling (ii) Grammar (iii) Readability (iv) Essay Length and (v) Concepts were chosen for the prototype implementation. The developed criteria will be described in more detail after the architectural overview of the system. To foster development existing libraries available for the Java platform where used when possible to implement the different criteria.

5.3. Architecture Overview

To meet the flexibility and extensibility requirements a plugin-based approach for the evaluation criteria is the best choice. This way it is easy to develop different criteria or try different versions of one criterion.

Figure 5.1 gives a schematic overview of the system. The system is split into several logical modules[2] for task separation. One design goal was to enable parallel evaluation processing for faster processing or load distribution, although this is not done by default. In the following sections each module of the architecture is described except for the plugin criteria. A detailed description of them is given in the implementation Section 5.5.

5.3.1. Database Abstraction

All operational data is stored in a database to ensure it is available across platforms and over the network. This design also allows multiple clients to run at the same time (e.g. students reviewing their results through a website and teachers grading another assignment). Structured Query Language (SQL) is used to access the data allowing different databases to be used locally and remote. The native JDBC Java API is used as abstraction layer for data manipulation. Although JDBC supports different database implementations if appropriate drivers are available, another abstraction for database

[1] New BSD License (Modified BSD License) or Simplified BSD License (FreeBSD License)

[2] the Java package layout differs slightly from the logical layout due to restrictions on access modifiers

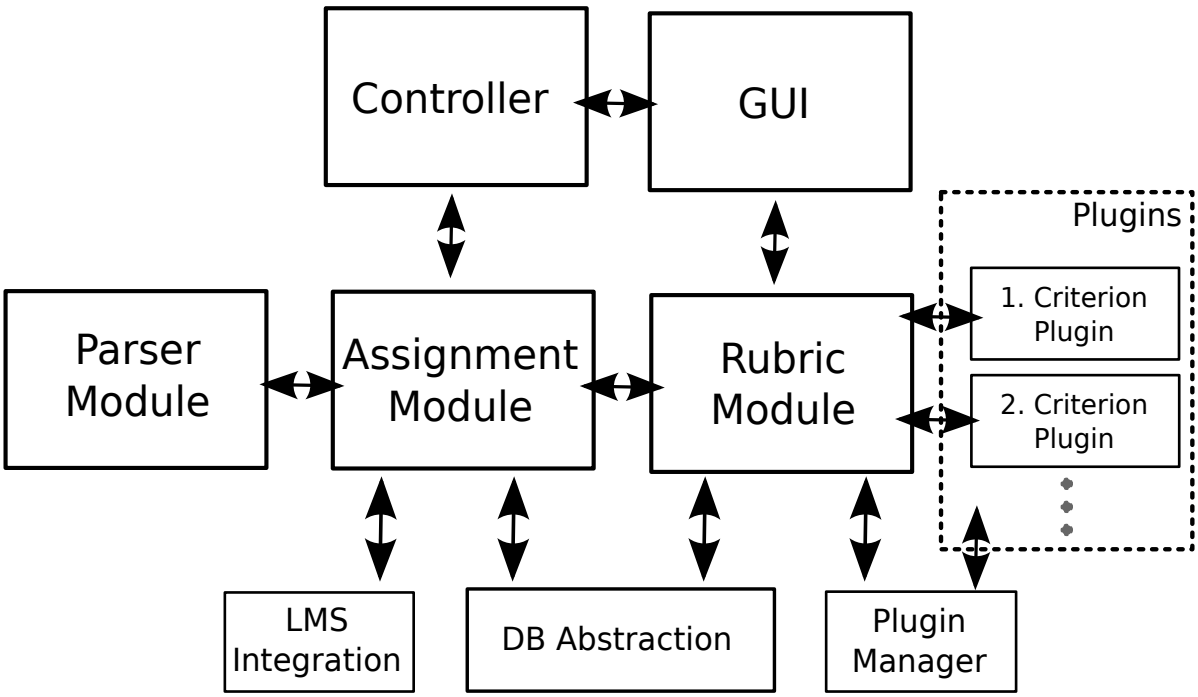


Figure 5.1.: System Architecture

providers was implemented to ease connection initialisation with different databases and to account for differing SQL dialects in table definition. The system expects the database to be empty and at the first run and will create all necessary tables automatically.

5.3.2. Assignment Module

The assignment module offers internal representation of an assignment with associated data like deadlines and the essay submissions and student relations. It is the connection point for a possible integration into a LMS. An integration would retrieve course and assignment details from the LMS along with the student submissions. To address privacy concerns, internally the only student related information stored is an id number matching submissions and results. The assignment module receives the student submissions and utilizes the parser module to store them in an internal XML representation along with the original submissions.

5.3.3. LMS integration

Implementations of the LMS integration interface provide course and assignment details along with the student submissions. They are especially responsible to translate the internal student id to the student identifier used in the LMS system and therefore may provide access to personal student data which is not stored locally.

5.3.4. Rubric Module

The rubric module provides a logical rubric abstraction which is assembled by the criteria. It is the core module providing most of the functions of the system. Criteria are implemented as plugins to enable easy extension of the system. Criteria must implement specified interfaces and certain parts must be derived from provided base classes to ensure a consistent behaviour of the system (refer to Section 5.5.4 for more details). The rubric module handles the evaluation of a submitted essay through triggering evaluation by each criterion and summarizing the individual results. It also handles manual ratings by teachers alongside the automatic ratings which are stored separately. Available criteria are discovered through the plugin manager.

5.3.5. Plugin Manager

The plugin manager utilizes a plugin framework to build a list of available plugins. A plugin can provide a single criterion or may provide several criteria at once. Two interfaces are defined for plugins to separate basic functionality from GUI integration. The GUI integration allows plugins to provide configuration screens in the system configuration. This can be used for example to specify the location of dictionaries for a spell-checker plugin.

5.3.6. GUI

The GUI is completely separated from actual data processing except for handling and possibly transforming user input data. Therefore it can be easily exchanged with different implementations. Different views must be easily managed within the GUI and so a docking window approach, as known from many Integrated Development Environments (IDEs) for programming, was chosen for the prototype implementation. This allows users to customize their workbench to the style they prefer and also supports the efficient usage of multiple displays as views can be placed on different displays. Relevant internal state data like the currently open assignment are stored in this module allowing to easily keep a consistent GUI state.

5.3.7. Parser Module

The parser module is responsible to translate the incoming student essays into the internal XML representation. After the format conversion basic processing functions as stemming and POS tagging are performed on each essay. The processed essays are stored in the internal database alongside with the original document. It is therefore possible to repeat the parsing at any time or allow specialised criteria implementation access to the raw essay data.

5.3.8. Controller Module

The controller module is responsible to initialize required system modules like the plugin manager. Secondly it is utilized by the GUI module through user interactions to perform high-level tasks as creating new assignments or triggering essay evaluation. These functions are provided separately to ease client development without knowing all system internals. No internal states except for system initialisation status are stored in this module. Therefore the module can be used by different client implementations at the same time.

5.4. Toolkits and Libraries

Several toolkits and libraries have been used to implement the prototype. In this Section only those utilized in the general architecture are described. Libraries specific to a criterion are named later in the detailed criteria descriptions.

5.4.1. Plugins

A library providing services like extension discovery and module loading eases the implementation of the plugin approach for criteria. The Java Plug-in Framework (JPF)[1] was chosen because it is a lightweight library providing the required extensibility through plugin management and it is licensed under the LGPL version 2.1. More advanced approaches as the OSGi framework[2] have been evaluated but have been considered as to heavy weight for a first prototype. Later implementations could probably benefit from the remote management services OSGi offers in deployment, especially in a Service-Oriented Architecture (SOA).

5.4.2. GUI

For the standalone prototype the InfoNode Docking Windows[3] library was used as the docking window implementation. The library provides the support to handle windows as views which can be flexibly placed side by side, as tabs or separate floating windows. Views can be easily resized and dragged by the user to change the program layout. User defined layouts can be stored and reused or the default layout being restored at any time. Last state restoring can be configured to automatically resume with the last edited assignment and essay to minimize workflow disruption in semi-automatic grading.

Several SWING extension libraries were used to provide better GUI interaction and rendering. MiGLayout[4] is used mostly as the component layout manager. The advantage of MiGLayout to comparable layout managers is that it supports SWING, SWT (Eclipse)

[1] `http://jpf.sourceforge.net`
[2] `http://www.osgi.org/About/Technology`
[3] `http://www.infonode.net/index.html?idw`
[4] `http://www.miglayout.com`

and JavaFX at the same time and is available under the GNU General Public License (GPL) as well as the Berkeley Software Distribution (BSD) license.

Additional SWING GUI components are provided by the SwingX [1], Flamingo[2], Cobra toolkit[3], Jxlayer[4] and Wizard[5] libraries. The used look and feel for rendering the components is Substance[6].

5.4.3. Text processing

While the submitted essays are parsed into the internal XML representation the texts are tagged with various additional attributes. Especially each word is tagged with the attributes POS, a stemmed version and possibly a baseform. For example the baseform for a noun as *networks* would be the singular form *network*. In some cases these will be identical to the original or stemmed version.

POS tagging is performed with the Stanford POSTagger[7] which is a log-linear part-of-speech tagger (Toutanova & Manning, 2000). The library is dual licensed under the GPL for research and open source software and under a commercial license for proprietary software. The tagger uses the Penn Treebank tag set as described by Marcus, Santorini, and Marcinkiewicz (1993) which is therefore used as the tag set in the prototype implementation. POS tagging algorithms generally operate on a whole sentence as minimal input and internally tokenize the sentence into words. Different strategies for tokenizing a sentence exist (e.g. Treebank homogenization[8]) yielding to slightly different results possibly requiring to match the found tokens which may be possible only partly in certain cases.

Stemming is performed with the Porter2 stemming algorithm for English (Porter, 2002). It is available as part of the BSD licensed Snowball stemming toolkit[9]. Snowball is a language to define stemming algorithms which can be translated to ANSI C and Java programms (Porter, 2001). Several stemming algorithms for languages as for example English, French, Spanish, Italian, German, Swedish are available including the original algorithm for English developed by Porter in 1980 known as the Porter stemmer (Willet, 2006). A newer version of the original algorithm sometimes referred to as Porter2 is the default stemmer for English in the toolkit (Porter, 2002).

[1] `http://java.net/projects/swingx`, license LGPL 2.1
[2] `http://java.net/projects/flamingo`, license BSD
[3] `http://lobobrowser.org/cobra.jsp`, license LGPL
[4] `http://java.net/projects/jxlayer`,license BSD
[5] `http://java.net/projects/wizard/`, license CDDL 1.0
[6] `http://java.net/projects/substance-swingx`, license LGPL 2.1
[7] `http://nlp.stanford.edu/software/tagger.shtml`
[8] `http://www.cis.upenn.edu/~treebank/tokenization.html`
[9] `http://snowball.tartarus.org`, license: `http://snowball.tartarus.org/license.php`

5.4.4. Database

Two database implementations are supported. First a local integrated database is provided through Apache Derby[1]. Derby is an SQL database which can be used as an internally embedded database as utilized in the prototype. Nevertheless it would support the more common client/server mode as well. It has build in support for encryption which can be used to achieve a higher privacy level preventing direct access to the database. Secondly the Connector/J[2] JDBC driver is included to access local and remote MySQL databases. The driver is dual licensed under the GPL 2.0 and a commercial license from Oracle Corporation[3].

5.5. Implementation

This Section explains certain design decisions and covers some internal aspects in more detail. References to used design patterns are included in the descriptions.

5.5.1. Parser Module

Different parsers are needed depending on the document input format. Therefore this part follows a creational pattern known as Factory Pattern (Gamma, Helm, Johnson, & Vlissides, 1995, p. 107). The concrete implementation is a parametrized factory method taking a file and creating an appropriate parser object according to the file-type. The factory itself is implemented as a Singleton (Gamma et al., 1995, p. 127) to ensure parser instances are only created once as this can be a rather heavyweight operation. Consequently the current implementation to import essays may not be thread-safe. The essays are basically parsed into a structure of paragraphs, sentences and words (see the following Section 5.5.2 and Appendix E).

5.5.2. Internal Essay Format

Essays imported into the system are initially parsed into an internal XML representation. The original and the XML representation of the document are stored in the database. The original file is only used to make the original formatting available for teachers when reviewing essays and optionally for specialised criterion algorithms. Regularly all evaluations are computed with the XML representation which supports annotating text parts. This is for example used to highlight parts of the text in the graphical view. For visual representation the XML document is transformed into an HTML document which is displayed in a SWING component (see Section 5.5.5 for further details). The internal format starts with some statistics about the text as for example the number of paragraphs. The text is parsed into the entities (i) "headings" (ii) "paragraphs" (iii) "sentences" and

[1] `http://db.apache.org/derby`, Apache 2.0 license
[2] `http://dev.mysql.com/downloads/connector/j/5.1.html`
[3] `http://www.oracle.com`

(iv) "words". Punctuation characters as full stops are also represented separately. Every entity gets a unique identifier assigned. Words are attributed with a lower-case, stemmed and baseform version. Refer to Appendix E for the exact XML format and an example parsed essay.

5.5.3. Plugins

Plugins are handled in the plugin manager concerning discovery and initialisation. Plugins are loaded when criteria they provide have been used in the currently selected rubric. Optionally users can configure the system to load all available plugins initially, which increases the system startup time but makes switching between rubrics faster. A valid plugin must:

- implement the *RubricAnalyserPlugin* interface
- may implement the *RubricAnalyserPluginGUI* interface
- provide at least one criterion implementation derived from the abstract *Criterion* base-class

The plugin interface specifies methods to retrieve instances from the implemented criteria and to deserialize stored instances. The contract for a plugin is that when it is loaded its initialisation method is called before any criteria instances are created. The optional GUI interface provides methods allowing the plugin to provide a configuration screen and possibly a main menu entry which is created when the plugin is loaded. Again the provided initialisation method for the GUI will be called during plugin loading. Refer to Appendix I for the interface specification.

5.5.4. Rubric Module

The rubric modules contains the rubric class and the base classes and interfaces for rubric criteria. The rubric class is a logical representation similar to a real paper rubric as it groups the used criteria together. The rubric assembles the overall result for an essay evaluation and delegates individual criterion evaluation to the implementing class. The design of the rubric module can therefore be seen to follow the Master-Slave pattern (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996, p. 245) as the task to calculate the result for a given essay is split into several sub-tasks assigned to the criterion implementations as slaves. The rubric assembles the individual criterion results and returns the overall summarized result to the calling client. The Master-Slave pattern allows concurrent processing leading to faster response times on multi-core systems.

Criteria Each criterion must be sub-classed from an abstract base class specifying the usage contract in the rubric module. Optionally a second interface can be implemented if the criterion provides its own GUI components. The docking window design allows the criteria to add complete views for own visualisation purposes if necessary. Summarized a valid criterion has to meet the following requirements:

- must be derived from the abstract *Criterion* base-class
- must implement the *Serializable* interface
- may implement the *CriterionGUI* interface

An abstract base-class was used for the criteria instead of an interface as criteria contain a common set of basic functionally which can be implemented in the base-class. For example setting and retrieving the weight, title or the reached level is the same for each criterion. Refer to Appendix I for the class and interface specifications.

It is possible that a criterion only offers processing functions but does not add own elements to the GUI. This is more of a logical separation as most criteria will implement both interfaces to be able to provide configuration options and analysis elements to the user.

5.5.5. GUI

The GUI follows largely the Model-View-Controller pattern (Buschmann et al., 1996, p. 125). SWING components themselves often follow the pattern but the concept is employed in the overall system architecture as well. Views are strictly separated from the application model. Controllers are mostly coded as input controllers through SWING action listeners either directly manipulating the views or deferring control to logical management classes in the model. To update the data in the views the system heavily uses the Observer pattern (Gamma et al., 1995, p. 293) where views register listeners which are notified in the case of data changes.

In Text Highlighting Criteria often need the possibility to highlight parts of the essay text for visual clues. Words, sentences and paragraphs are possible targets to be visually marked. Textual highlighting is implemented by adding classes to the appropriate text parts and adding Cascading Style Sheets (CSS) rules accordingly. Depending on which CSS rules are activated in the XML to HTML transformation different text parts can be efficiently highlighted. The transformation is done with an Extensible Stylesheet Language (XSL) stylesheet configuration file stored in the configuration directory. The used XSL stylesheet can be found also in Appendix F.

This solution is transparent to the evaluation process only affecting visual representation clearly separating data processing and visualization. It is however heavily depending on the final format therefore limiting output transformation to the HTML format. Although the support for CSS of the primary GUI target is limited the necessary basic highlighting could be realized with it. As a webclient is a logical further system extension (see non-functional requirements 5.2.1), this decision actively supports the development of an alternative webclient GUI. Therefore the CSS highlighting approach was judged appropriate for the prototype implementation.

5.5.6. Testing

During development unit tests were developed for all non-GUI system parts. Unit testing is essentially helpful when parts of the system are refactored during development catching

errors introduced through these changes. The benefits of the automatic testing greatly outweigh the effort needed to adapt the test cases to system changes.

It takes far more effort to automatize GUI testing which was not implemented in the prototype system but a manual testing procedure was applied. Basis for manual GUI testing are the tasks a user has to accomplish with the system. Each time a non-obvious bug in the GUI was found a protocol of the steps to repeat the error was written down. This way a collection of GUI tests was gradually build up which was used after re-factoring GUI parts to test proper functionality.

Aside from the collection of task-based GUI tests heuristic evaluations (Molich & Nielsen, 1990) have been done during the development cycle. The used set of heuristics followed the published user interface heuristics by Nielson (1993, p. 115-155).

5.5.7. Spell-checking Criterion

Depending on the used dictionary computers perform quite well in spell-checking. Not many open source or adequately licensed spell-checking libraries for Java exist reducing the available options. To implement a spelling criterion the open source LGPL [1] licensed Jazzy[2] library was finally chosen. According to the documentation it uses an algorithms similar to GNU Aspell[3]. White (2004) gives a short review of Jazzy also explaining general algorithms used in spell-checker engines being phonetic matching and sequence comparison algorithms. As Jazzy is based on Aspell it uses the metaphone[4] algorithm for phonetic encoding (P. E. Black, 2007) and a variation of the Levenshtein algorithm (White, 2004). The usage of the spell-checking engine as a rubric criterion differs from the usage in text processing software as correction suggestions are not important and only errors are counted and marked. For each level in the rubric the maximum amount of acceptable errors to still reach the level is specified by the teacher. It is suggested to still accept a few errors at the highest level to account for the problem with false positives from words missing in the used dictionary.

The English dictionary used in the plugin implementation is from SCOWL (Spell Checker Oriented Word Lists)[5] and was generated with the included `mk-list` script with the options `english 80`. General dictionaries are well suited for prose but detection of errors may result in a high rate of false positives in text containing a lot of domain specific terms. To reduce the number of false positives the example essays attached to the rubric are scanned for errors. When the criterion is configured a list of spelling errors in these example essays is presented and all false positives can be marked as correct. These are used as a white-list in further essay evaluation. The list is stored specific to the rubric for long term reliability of the results. A user specific dictionary in addition to the default dictionary would change over time. Therefore it would be impossible to reliably recalculate the results for an essay at a later point in time.

[1] license version 2.1.1, `http://www.gnu.org/licenses/lgpl-2.1.txt`

[2] `http://jazzy.sourceforge.net`, Version 0.5.1

[3] `http://aspell.net/man-html/`

[4] `http://aspell.net/metaphone/`

[5] `http://wordlist.sourceforge.net/`

Improvements The current primitive evaluation algorithm is language independent allowing to add different dictionaries for other languages. If this feature should be used in a future version the included phonetic matching algorithm of the spell-checking library should be adapted to the language. Support to use different dictionaries for one language at the same time should be added. As outlined in the white-list problem above an evaluation at a later point in time should yield the same results. The current implementation assumes the default dictionary will not change. This assumption is erroneous as over time dictionaries are updated and the included dictionary can be changed by the user. Therefore the used dictionary should be backed up with the rubric to reliably recalculate the automatic rating at any point in time. Avoiding false positives would reduce human effort further. One possibility for this would be context-sensitive spell-checking algorithms as WinSpell which is based on a Winnow algorithm (Golding & Roth, 1999). WinSpell and similarly BaySpell require a training set of data. The example essays attached to the rubric can be used to tune such an algorithm, but as these will be only a few this might not be sufficient as a full training set for such an algorithm. Raising the amount of required example essays would contradict the goal to reduce human effort and must therefore be strictly avoided.

5.5.8. Grammar Criterion

Far fewer freely usable grammar checkers exist compared to spell-checkers. Grammar checkers are even more language dependent than spell-checkers as they need to analyse sentence structure and not only compare words. LanguageTool[1] is an open-source LGPL licensed grammar checker capable of handling different languages as English, German, Polish, Dutch, French, Romanian, Italian and Russian and initial support for further languages (Miłkowski, 2010). The first version of the tool was developed in Python as a rule based grammar checker (Naber, 2003). It was later ported to Java with the target of integration into OpenOffice as an extension but can also be integrated into other applications or used as a server-side process (Miłkowski, 2010). New rules can be easily added through XML configuration files and more complex rules can be developed as Java classes extending the basic Rule class. The expandability and coverage of several language makes LanguageTool a valid choice to be integrated as a rubric criterion.

The library is directly integrated as a criterion counting the grammatical errors in the essays. Evaluation follows the spell-checker principle of definition of a maximum error count per rubric level. Although grammar rules try to avoid false positives with a more lenient checking, missing possibly errors, they can still occur and should be addressed in the set level limits. The current implementation does not benefit form the example essays added to the rubric as new rules cannot be detected automatically from the example essays.

Improvements In the current integration only the English rules are active. As the library supports more languages these should be included as well. The same problem as

[1] `http://www.languagetool.org`

in spell-checking with updated dictionaries applies to the rules used by the grammar checker. Rules should therefore not be updated without backup of the previous version. LanguageTool is not the best available grammar checker compared to commercial solutions but it can be easily improved with the development of new rules. Through the plugin design it is easy to add different grammar checkers which might perform better on certain essays.

5.5.9. Readability Criterion

Different formulas to calculate the readability of a text exists. Rudolf Flesch published a readability formula in the article "A New Readability Yardstick" in the Journal of applied Psychology in 1948 called the Flesch Reading Ease formula (as cited in (DuBay, 2004)) based on words, sentences and syllables counts. The score is calculated according to the following formula (Flesch, 1979):

$$206.835 - 1.015\frac{\text{total words}}{\text{total sentences}} - 84.6\frac{\text{total syllables}}{\text{total words}}$$

The formula was chosen as it is still widely used. For example its use is mandatory to check the readability of insurance policies in Florida (*FL Statutes - Title XXXVII Insurance Section 627.011*, 2004). Syllables are counted with the Morphadorner library[1]. Refer to table 5.1 to interpret the resulting score of the formula. A revised version of the formula called Flesch-Kincaid formula or Flesch-Grade Level was developed to match the score with U.S. grade levels (DuBay, 2004; Wikipedia, 2010a; Dragan & Woo, 2010).

Improvements Add different formulas to choose when configuring the algorithm including short explanations for the teachers. As the Flesch Reading Ease formula was already published in 1948 newer formulas delivering better results might be available.

Table 5.1.: Flesch's Reading Ease Scores

Reading Ease Score	Style Description	Estimated Reading Grade	Estimated Percent of U.S. Adults (1949)
0 to 30	Very Difficult	College graduates	4.5
30 to 40	Difficult	13th to 16th grade	33
50 to 60	Fairly Difficult	10th to 12th grade	54
60 to 70	Standard	8th to 9th grade	83
70 to 80	Fairly Easy	7th grade	88
80 to 90	Easy	6th grade	91
90 to 100	Very Easy	5th grade	93

Flesch's Reading Ease Scores published in "The Art of Readable Writing", Flesch (1949, p. 149)
as cited in (DuBay, 2004, Table 3)

[1] `http://morphadorner.northwestern.edu/morphadorner/licenses`

5.5.10. Essay Length Criterion

Sometimes teachers want students to achieve a specific essay length for an assignment. Essay length can be best measured as a word count. This number is independent from the formatting of the text (contrasting to page numbers which can be different due to font size, margins and paragraphing and are therefore not as suitable as a scale). This is an extremely easy check for a computer but rather exhausting for a human grader who would therefore not really count words but rather judge essay length through an optical overview on a printout. An expected essay length is usually given as an average length and not as an absolute limit. The implemented criterion can either take an explicit number as average text length or the average length of example essays from the rubric. Different levels are defined by the maximum percentage of derivation from the desired length to reach the specific level. This effects both essays longer and shorter than the the set average length.

5.5.11. Concepts Criterion

The concept criterion utilizes a web-service to discover the concepts covered in the example essays. Weinhofer (2010) describes the implemented concept extraction algorithm in detail. The algorithm was also used in the Enhanced Automatic Question Creator (EAQC) to extract *"the most important concepts"* and create *"single choice, multiple-choice, completion exercises and open ended questions on the basis of these concepts"* (Gütl, Lankmayr, Weinhofer, & Hofler, 2011; Gütl, Lankmayr, & Weinhofer, 2010).

The web-service delivers an XML document listing the found concepts. Each concept consists of one or several words defining the concept. For example the found concepts could be "networks", "computers", "physical objects" and "small scale copying". The returned concept definitions can be found literally, except for upper and lower case variations, in the source text send to the web-service. Teachers can complement the automatically found concepts by defining further concepts in the same manner. This results in a specification of the required concepts for a particular criterion instance in a rubric for a specific assignment. Furthermore the concepts are ranked according to the importance and therefore the expected coverage in the essays.

Evaluation of a student submission is done by computing an internal score expressing if the required concepts were covered in the essay. Because of the possibility in languages to express the same concept in different terms the web-service cannot be used alone to determine the covered concepts in the student essays to match the defined concept lists. To match the student essay content with the required concepts a large lexical database of English called WordNet® Version 3 is used (Miller, 1995). The database is freely available from Princeton University under a license[1] allowing commercial use.

The database groups words expressing the same concept into *synsets* and therefore can be used to discover matches in student essays using different expressions for the required concepts. As the number of found matches is still rather low the score is raised if a word in the student essay is a *hypernyms* of a required concept. The approach is not

[1] `http://wordnet.princeton.edu/wordnet/license/`

completely accurate as a word can have different senses which cannot be automatically determined from the specification of required concepts. Therefore a match is counted if an essay word matches any of the senses found for the words specifying a required concept.

To access the database which is provided in several flat text files a number of libraries exists to directly use the database files or to access a transformed version in an SQL database. For a first test of the approach the RiTA.WordNet[1] library was used. As the whole process can be quite demanding a performance comparison of the existing libraries should be done to improve system performance in future development.

Improvements According to Snow, Prakash, Jurafsky, and Ng (2007) "WordNet senses are too fine-grained" for many applications. Based on the outlined algorithm in the article, versions of the database with a different number of sense clusters are available from the Stanford Wordnet Project website[2]. Using these databases should yield a higher performance of the criterion evaluation algorithm as fewer synsets need to be compared. While the performance gains should be easily measurable a test design needs to be developed to determine with which database the algorithms yield more accurate results.

If overall system performance allows further processing, another possibility to discover concept instances in the student is to process the misspelled words. Generally a misspelled word will not be found in the WordNet database and therefore cannot contribute to the overall concept score. With a spell-checking engine delivering correction suggestions the most likely corrected word version could be used to query the WordNet database to determine the synset of the word. Due to the possibility that the system assumption of the most likely corrected word is wrong, a concept match could be found erratically. Therefore the internal score for such a concept match should be lower than a match from a not corrected word.

Retrieving the concepts out of the example essays is not always sufficient. It can be very helpful to be able to analyse another set of documents which are not necessarily essays like the set of literature used in a lecture. Reiterer, Dreher, and Gütl (2010) have developed a prototype to automatically retrieve concepts and relations between them out of a set of documents and present the found data as a "*radial space filling tree*". This visualisation allows to represent a hierarchical ontology in a condensed layout. This work could be integrated to assist teachers further in defining the concepts for the category. It can be used as a second source of concept suggestions, based on a broader source document set than the example essays. Secondly, if the prototype could be applied to a single essay as input data, the visualisation could be added to provide teachers with a graphical visualisation of the content of the student essay they are currently reviewing.

[1] `http://rednoise.org/rita/wordnet/documentation/index.htm`

[2] `http://ai.stanford.edu/~rion/swn/`

5.6. Summary

Rubric criteria should be logically independent from each other and therefore a plugin based approach can be followed fostering the fulfilment of the extensibility and flexibility requirements as outlined in Section 4.3. This also allowed parallel evaluation as a design constraint. The current design allows parallel evaluation of different essays as well as parallel evaluation of different criteria on the same essay. The latter are theoretically completely supported by the design but have not been tested extensively therefore leaving the potential of undetected synchronisation problems. Note that only evaluation is designed for concurrent use but for example initial document parsing when importing submissions is not thread safe in the prototype implementation.

For easier testing of the prototype Apache Derby[1] was included as a local database along with a simple installer to set up a local test machine speedily. As database initialisation differs between the local Derby database and a remote database like MySQL[2] a simple database provider abstraction was defined.

Although the system design allows some efficient parallel computations speed was not considered to be highly critical in the automatic evaluation phase due to the intended application at the classroom level. As the number of essays is limited, more processing time for a single essay can be used. Initial automatic evaluation may take a while which is considered acceptable as long as the further interaction in semi-automatic grading is reasonable fast for teachers using the system.

The developed prototype offers an easy and quickly set up environment to test different essay evaluation algorithms in a rubric implementation. The next Chapter will introduce the prototype and its usage.

[1] http://db.apache.org/derby
[2] http://dev.mysql.com

6. Usage and Evaluation

This Chapter starts with a walk-through of the prototype system. The tasks a teacher would perform with the system are explained with commented screenshots of the system. The second part is a consolidated evaluation of the prototype system.

6.1. Usage Instructions and Screenshots

A complete assessment consists of the phases (i) Preparation (ii) Rubric definition (iii) Rubric evaluation (iv) Student assignment submission period (v) Grading and (vi) Result publishing.

Teachers start with the assignment preparation which includes defining the assignment topic, the prompt for the students and deadlines. This phase is only basically covered by the prototype through entering the deadlines for an assignment. The next steps will be covered in the walk-through except for the result publishing step. The prototype implementation simply locks the results and prevents any changes when the publishing deadline is reached. At this point in a real system, the results would be either automatically published to the students or be exported into an LMS, which handles the actual publication.

6.1.1. Installation and Startup

The prototype includes a simple installation wizard (see Figure 6.1) which creates a local test instance of the protoype with an integrated database. This setup could be used by several users but not simultaneously as the internal database cannot be shared.

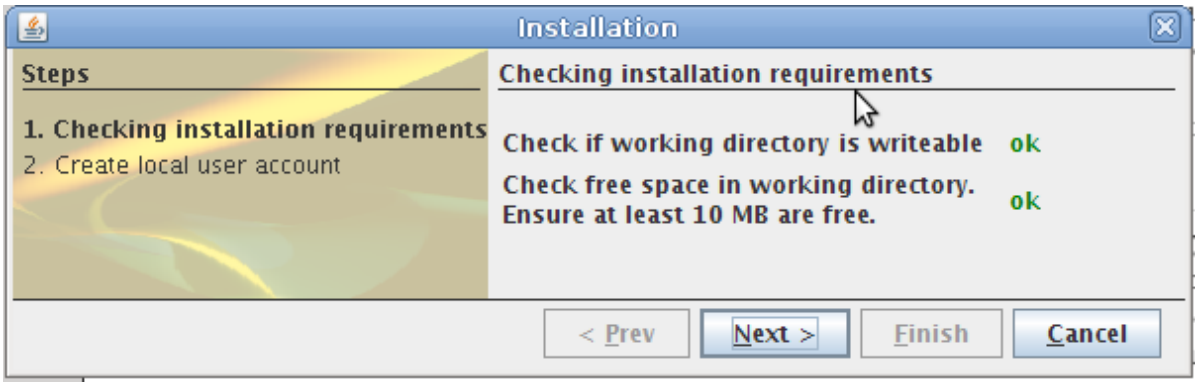


Figure 6.1.: Prototype Installation Screenshot

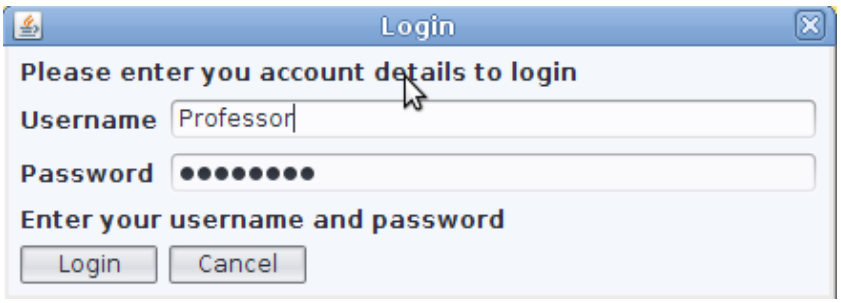


Figure 6.2.: Prototype Login screenshot

To account for a real usage scenario the user has to log into the prototype at startup (see Figure 6.2) as it would be the case in a deployed version using a shared central database. The login is also required to later support multiple raters for a single assignment.

If the application is run for the first time and no assignment has been created, a wizard is automatically opened. The wizard starts with condensed usage instructions (see Figure 6.3) and continuous with the specification of assignment details like a title and the deadlines. At the end, it is optionally possible to import the first batch of student submissions for this assignment.

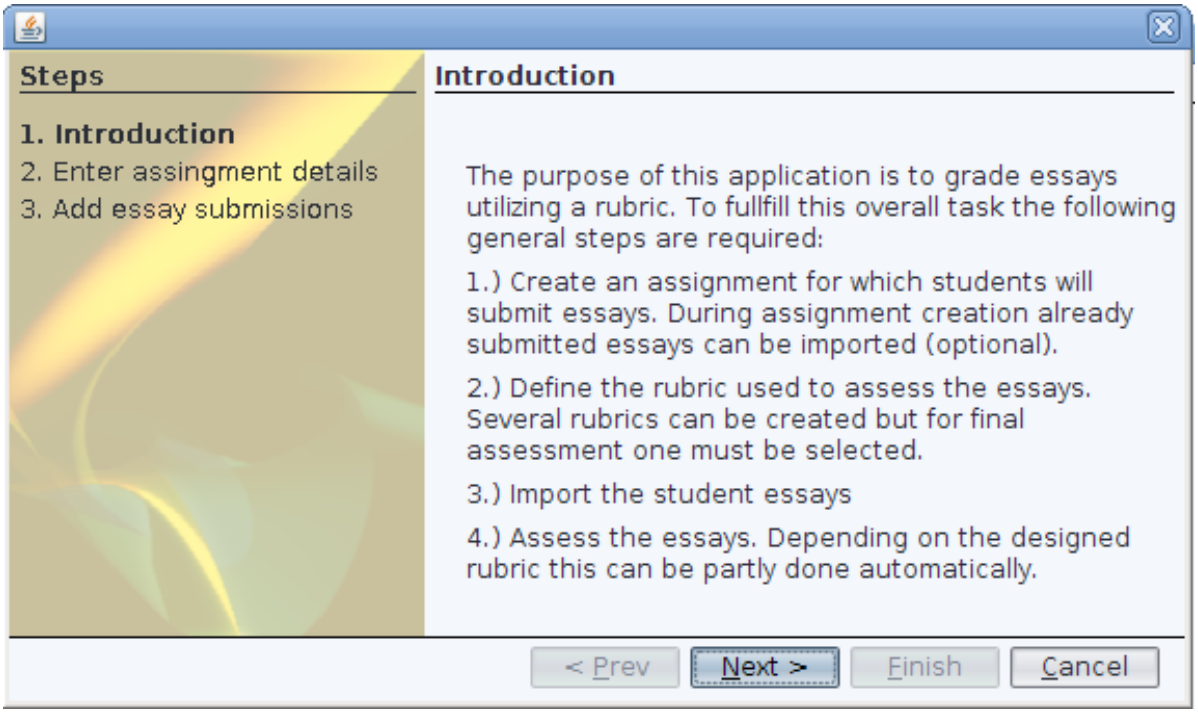


Figure 6.3.: Prototype First Assignment Wizard screenshot

6.1.2. Main Screen

The application main screen is a combination of different views (see Figure 6.4). On top the main functions are easily accessible with image buttons. All image buttons include an action text below them for usability reasons to counteract possibly differing symbol interpretations by users. Action linked directly to a single view are included at the top of the respective view. General functions not tied to a specific view are placed at the top. This is done due to the IDE window layout which allows to move views around or even put them in separate windows. The button placement ensure the appropriate actions are still reachable directly in nearly any layout.

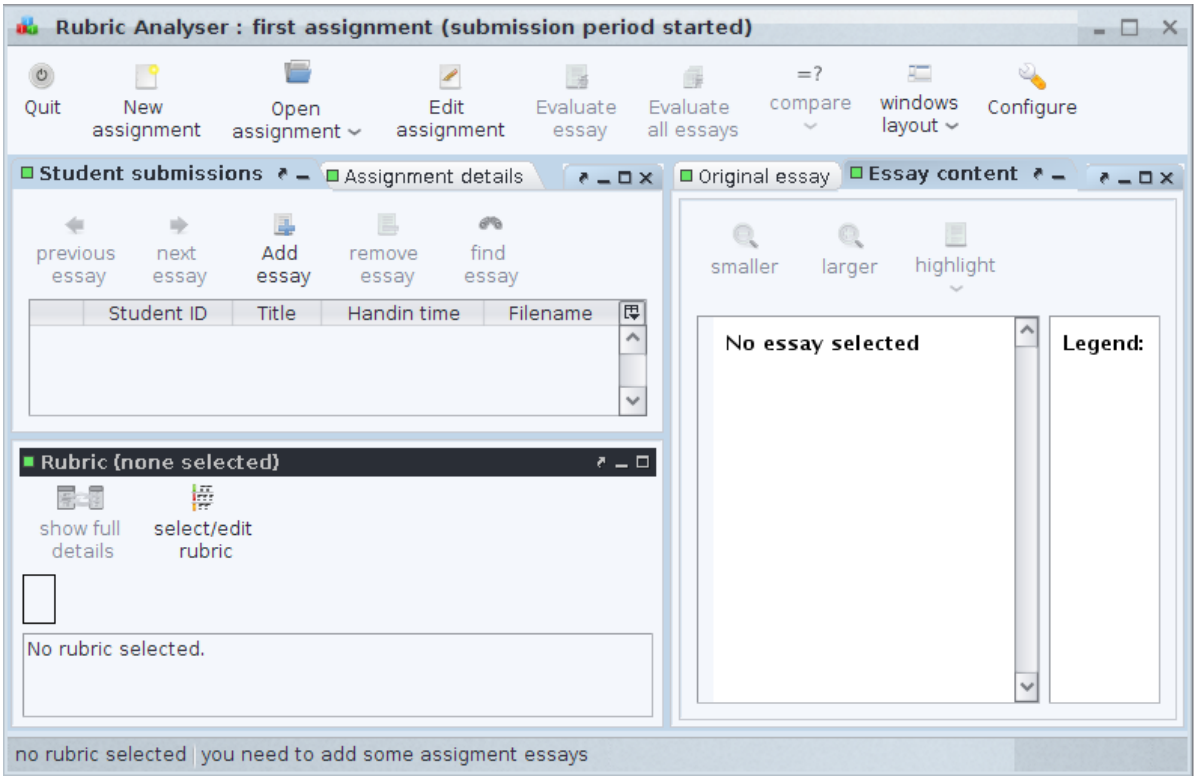


Figure 6.4.: Prototype Main Screen

The three main views are the essay list consisting of the student submissions, the current rubric and the currently selected essay. Additionally a view summarizing the assignment details is provided but hidden behind a tab in the default layout to save screen space. Two different essay views are provided, were again one is hidden behind a tab in the default layout. The main essay view is based on the internal XML format which is rendered as an HTML page within the GUI (see Figure 6.5). As this view is based on parsed data the formatting differs from the original one and is optimized for readability while grading. The original essay formatting is available in a second view as long as the original document format is supported for rendering (see Figure 6.6).

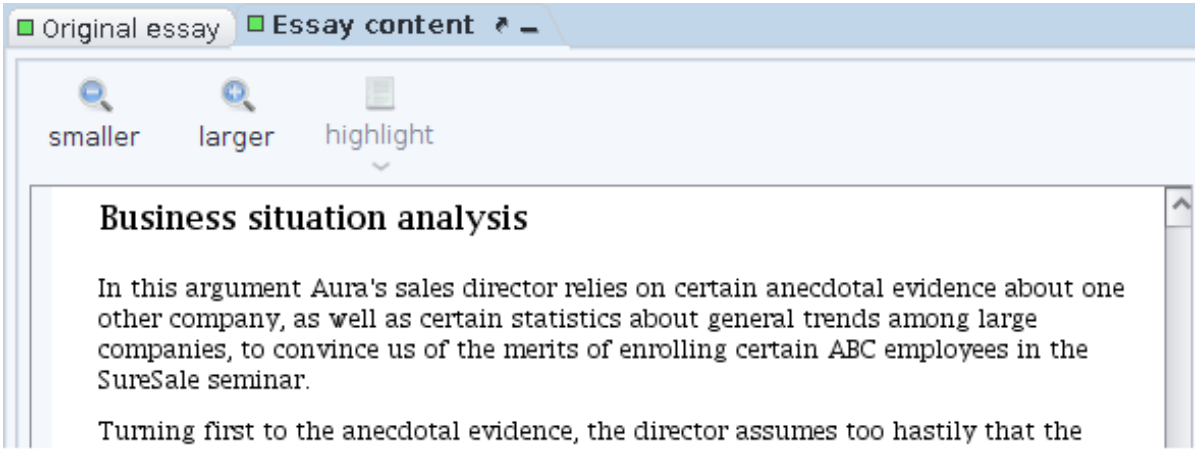


Figure 6.5.: Prototype Essay View

Users can adapt the main screen to their personal likings. Changed layouts can be explicitly stored and reused. By default the system stores the last configuration which will be restored at the next application run. In Figure 6.7 the rubric view was removed from the main window and is now a separate floating window above the remaining default main screen. This enables efficient usage of multiple monitor setups where users can place views on different monitors for an enhanced overview.

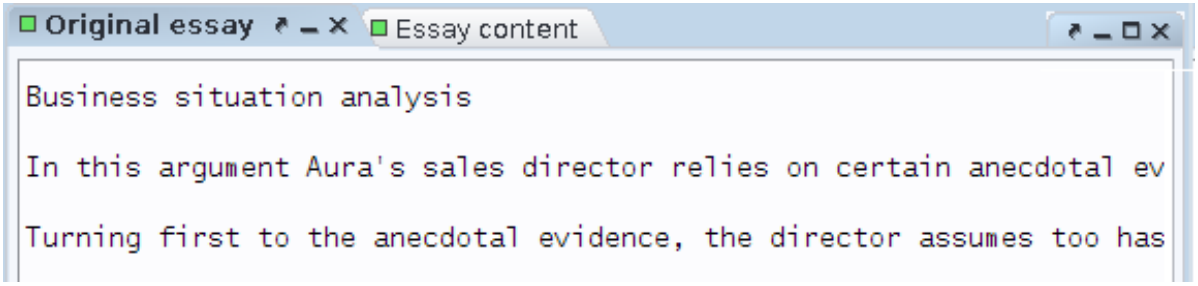


Figure 6.6.: Prototype Essay View Original Format

Nearly all interaction elements have short popup help screens similar to regular textual tooltips which are displayed when the cursor remains still over them. The popups are enhanced compared to the regular text tooltips as they allow several formatting options to display longer tooltips in a readable and consistent layout. Some GUI items have been adapted to better fit into the application. This is not exposed to the user, except for custom rendering, as the elements still exhibit the known behaviour.

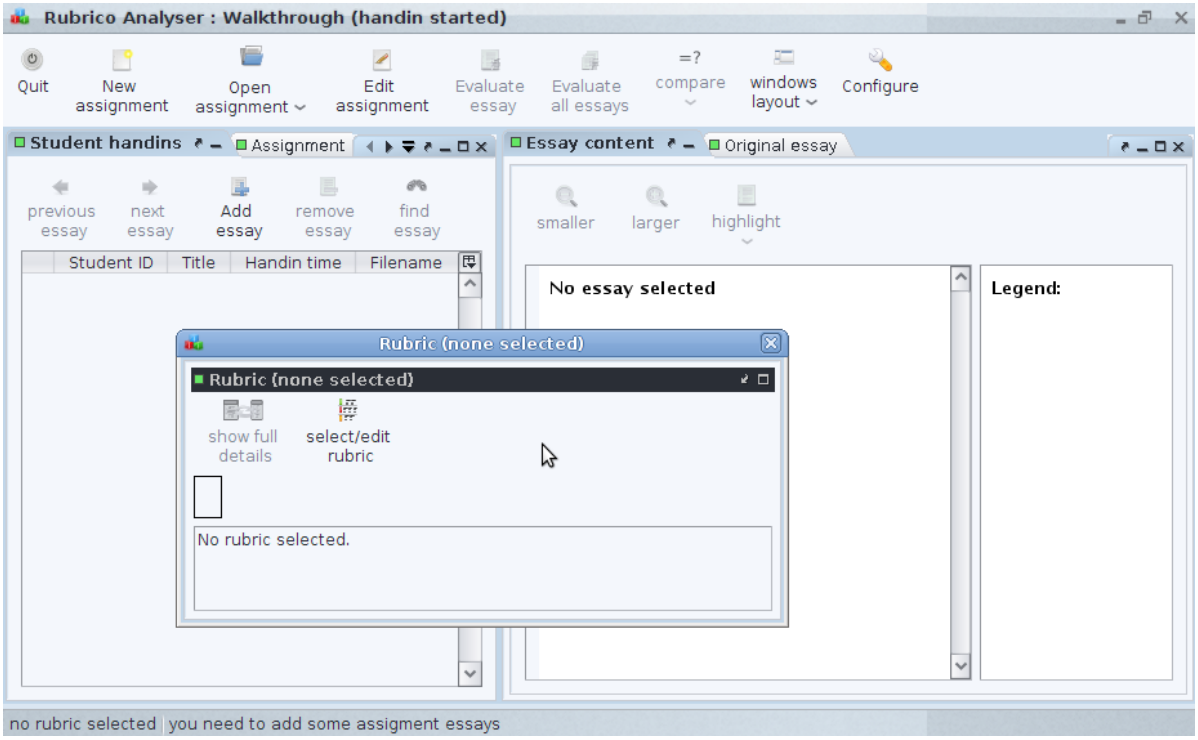


Figure 6.7.: Prototype Main Screen Floating View

Multiple views and windows can easily be confusing for users especially in cases where a dialogue or similar window is displayed on top which requires user interaction. The solution implemented for such cases is a spotlight approach as seen in Figure 6.8. Parts of the GUI currently not accessible are greyed out, focusing user attention to the current dialogue waiting for input. Similarly user input is visually blocked while longer processing tasks preventing further user input are carried out. In this case the screen is covered by a semi-transparent white overlay containing a spinning animation.

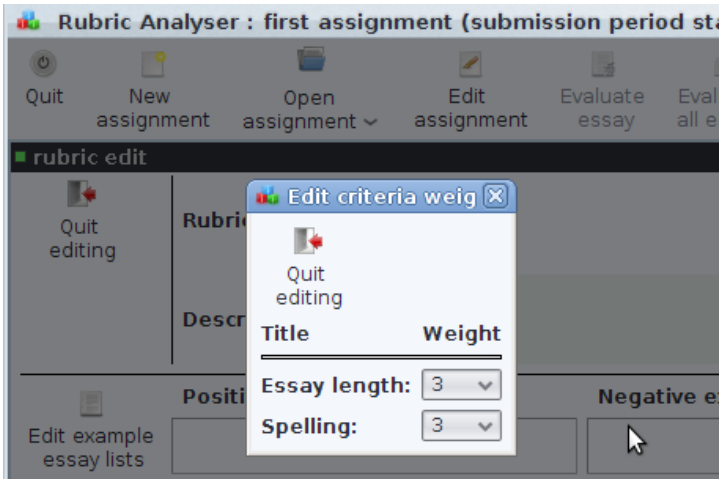


Figure 6.8.: Prototype Window Interaction Spotlight

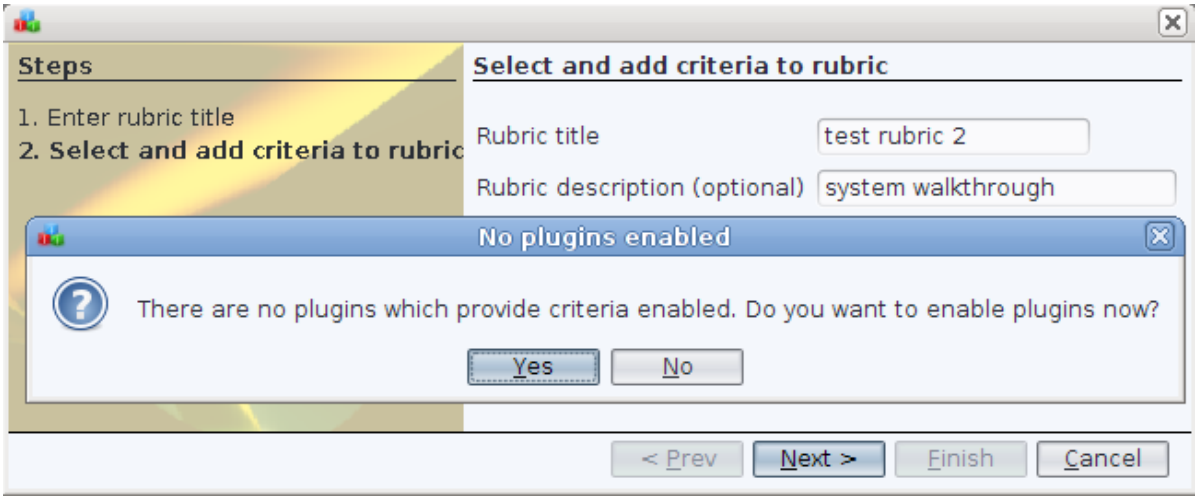


Figure 6.9.: Prototype New Rubric Plugin Activation

6.1.3. Rubric Definition

Defining the rubric used for an assignment is the most important preparation for later essay evaluation. For each assignment several rubrics can be defined but, only one is active at a time. The possibility to create more than one rubric per assignment is mainly useful to test different rubrics. For final essay evaluation one must be selected and should not be changed any more.

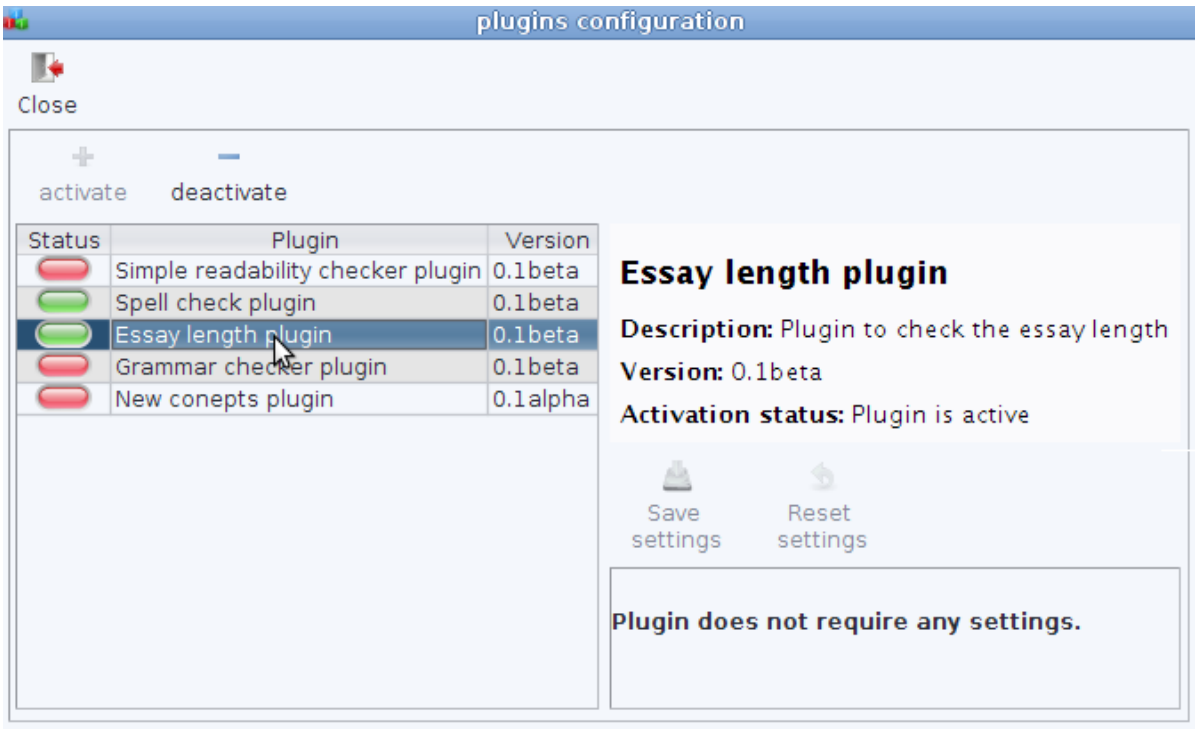


Figure 6.10.: Prototype Plugins Configuration

To create a new rubric the select/edit rubric button brings up a list of defined rubrics for the current assignment, also allowing to add a new rubric. This starts a wizard which asks for a title and optional description and allows to select the criteria to use for the rubric. As seen in Figure 6.9 the system asks the user to activate more plugins if no unused criteria are currently available.

Plugins might offer a configuration screen in the settings view allowing users to adapt the default behaviour of the criteria the plugin provides or specify necessary resources like for example the location of used dictionaries (see Figure 6.10). After the user activated plugins providing criteria, a selection is made for the new rubric and the edit view for the newly created rubric opens.

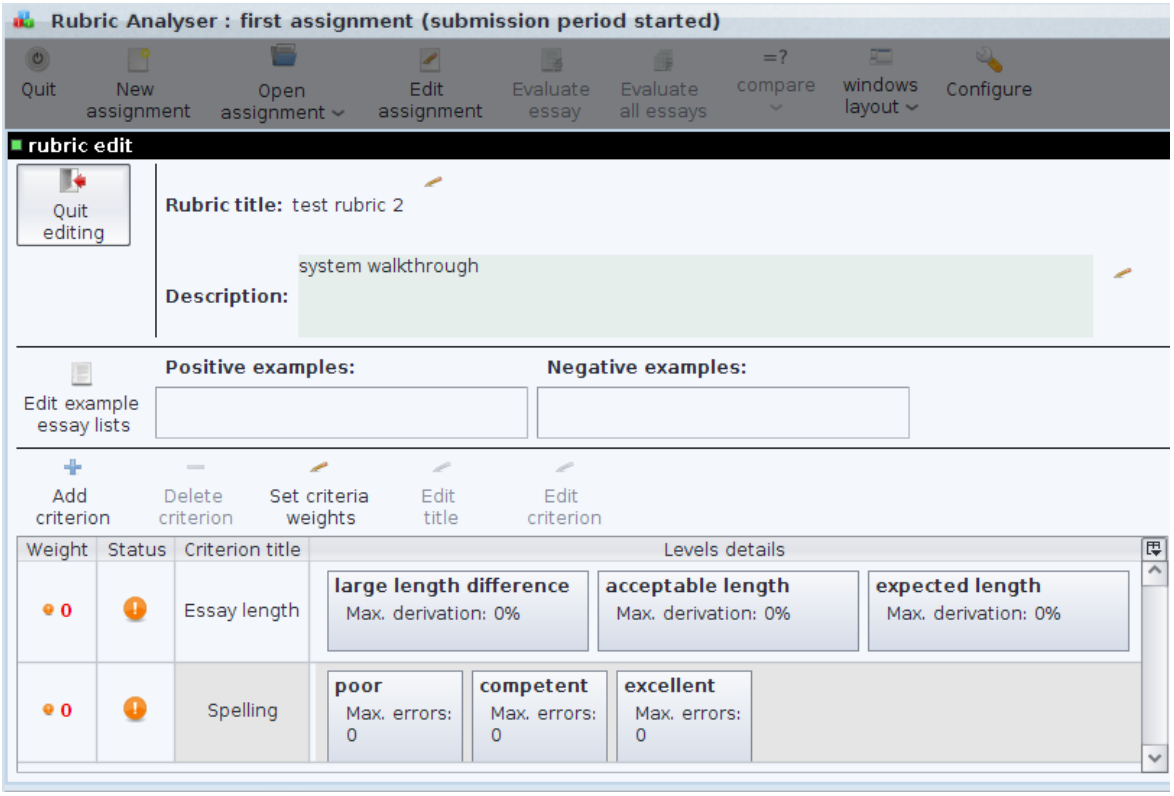


Figure 6.11.: Prototype Rubric Edit

Figure 6.11 shows the rubric edit view. The configuration status for each criterion is displayed. It is possible to quit rubric editing any time but only completely defined rubrics can be applied to the assignment. In the rubric edit view two main tasks have to be done: The weight for each criterion, which is used to calculate the overall score, has to be set (see Figure 6.12) and each criterion needs be configured.

Criteria definition views are provided through the respective plugin and therefore differ greatly. Common between all criteria is the definition of levels for each criterion which can range from 1 to 5 levels. Each criterion provides a default title which can be changed by teachers.

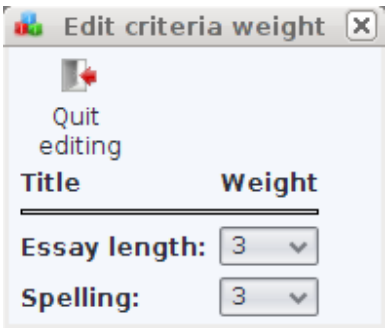


Figure 6.12.: Prototype Rubric Edit Criteria Weights

For each rubric it is possible to add positive and negative example essays (see Figure 6.13). Positive examples are considered as the standard solution for the prompt while negative examples are either insufficient or expected off-topic essays related to the prompt. The general rubric framework does not use these example essays but the criteria implementations can use them to configure their evaluation algorithms, This can either be done automatically or by supporting teachers in criteria configuration.. For example the essay length criterion will calculate the average essay length of the positive examples which can then be defined as the expected standard solution length for the student essays as seen in Figure 6.14. This figure is also an example for a criterion configuration screen provided by the criterion plugin implementation. Depending on the criterion a level definition must be specified so the algorithm can determine the reached level for a student essay. It is recommended that teachers apply the rubric on test essays to discover discrepancies between their expectations and the actual automatic results and so can adapt criteria configurations when necessary.

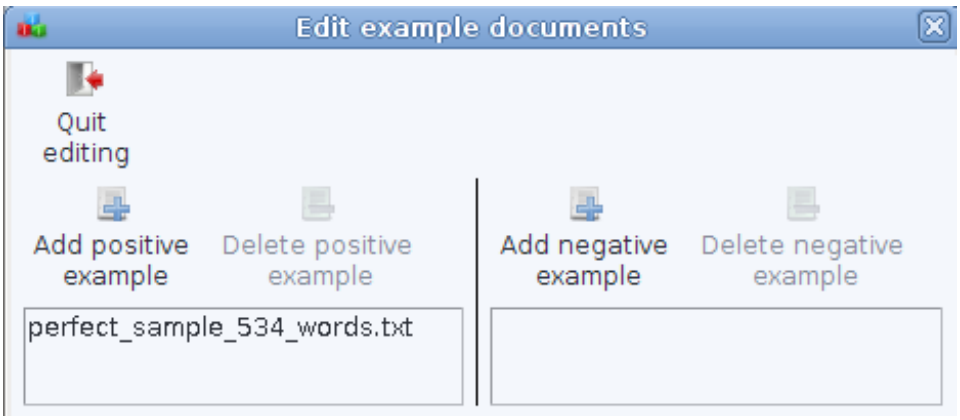


Figure 6.13.: Prototype Example Essays

Rubrics are by default viewed in a condensed form as seen in Figure 6.15. For each criterion only the reached level is displayed if the currently selected essay has been evaluated. Else an appropriate message is displayed instead of the actual level. Contrasting to the condensed view is the full view as seen in Figure 6.16 which is a familiar rubric representation as known from paper-based tabular layouts.

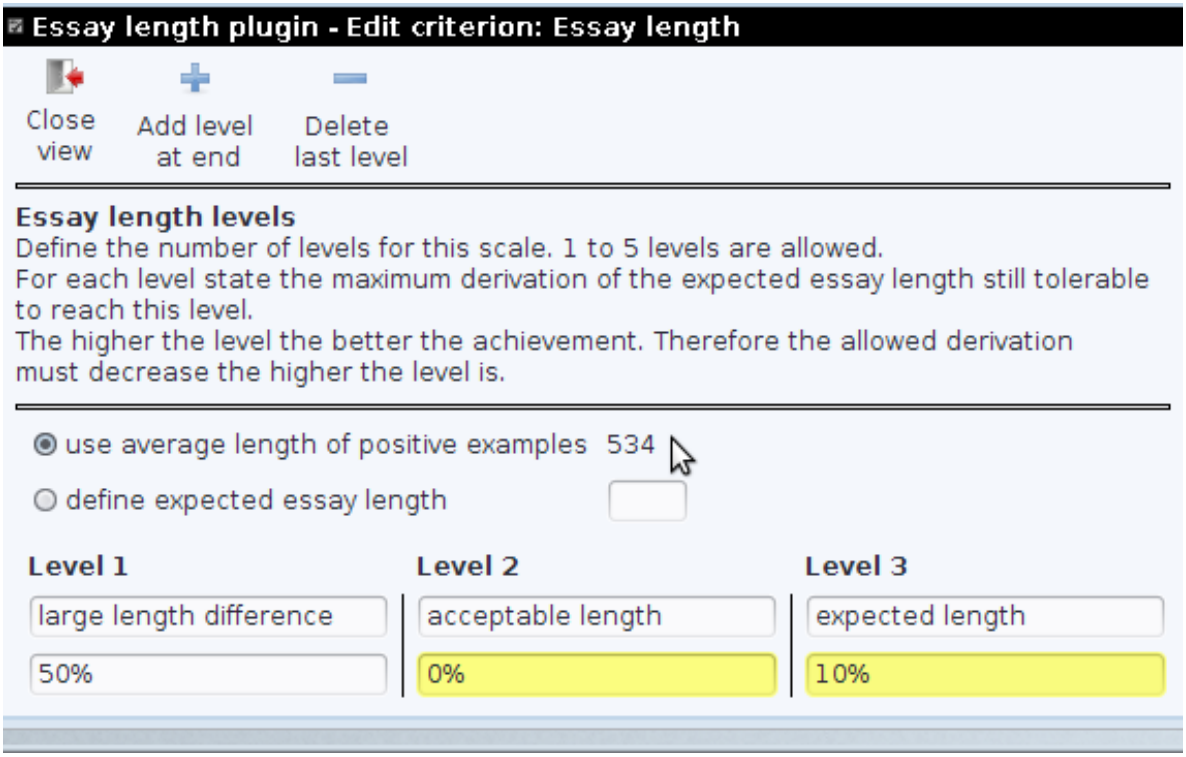


Figure 6.14.: Prototype Criterion Editing

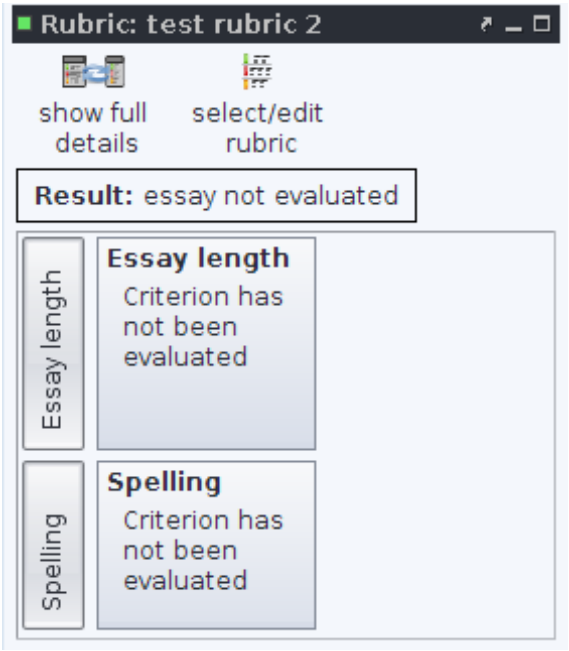


Figure 6.15.: Prototype Rubric Default View

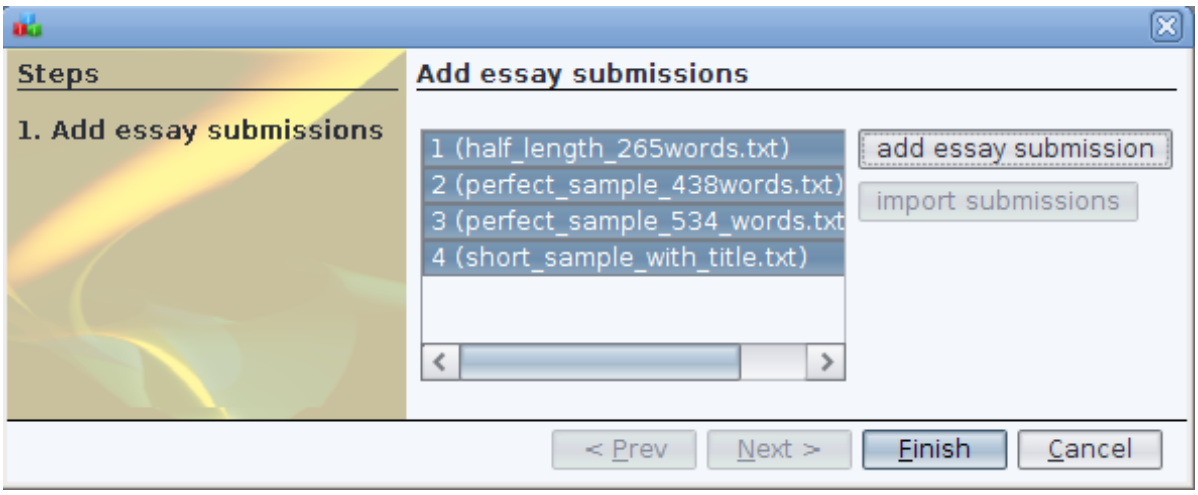


Figure 6.17.: Prototype Import Essays

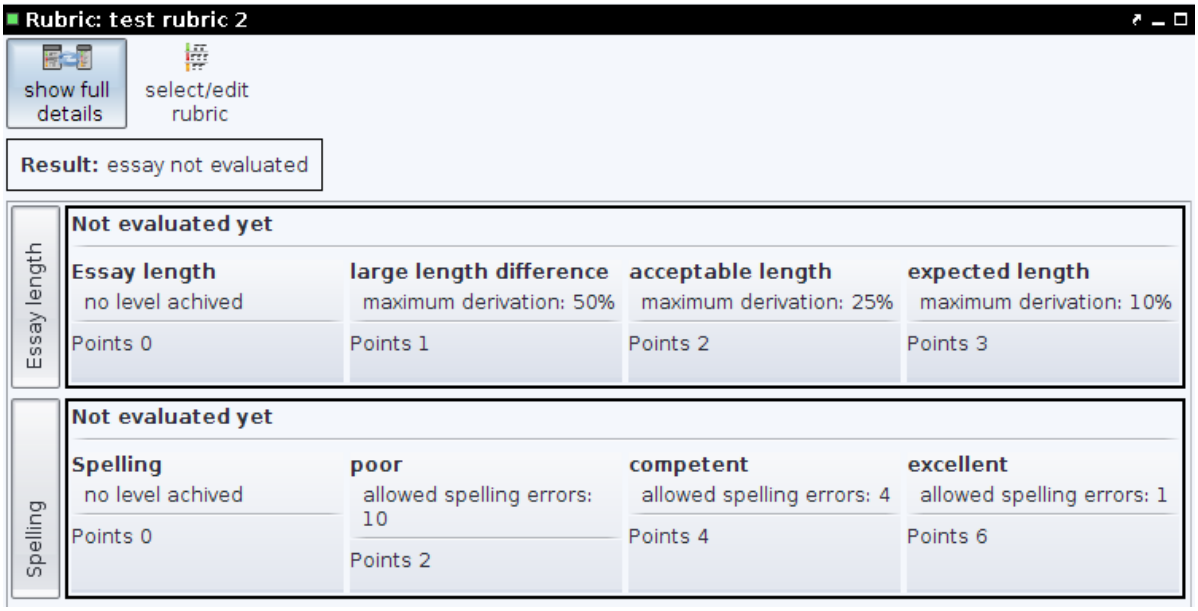


Figure 6.16.: Prototype Rubric Full View

6.1.4. Essay Evaluation

Before student essays can be evaluated they need to be imported into the system. In the prototype system this is a manual process as shown in Figure 6.17. In a improved system version this manual step would be replaced by an LMS integration providing automatic import of student essays. When student essays are available users can trigger automatic evaluation of all essays or only the currently selected one. This is useful to check if the rubric is working properly and before the lengthy operation of evaluation all essays is triggered.

After an automatic evaluation has been performed highlighters for text features are

available. Which are available depends on the used criteria actually providing the different highlighters. Figure 6.18 shows the highlighter provided by the spelling criterion. On the right side of the essay view a legend is automatically build using the same text formatting as used in the essay view for highlighting. There is no support to configure the used colours by users but this could be added in the system configuration screens.

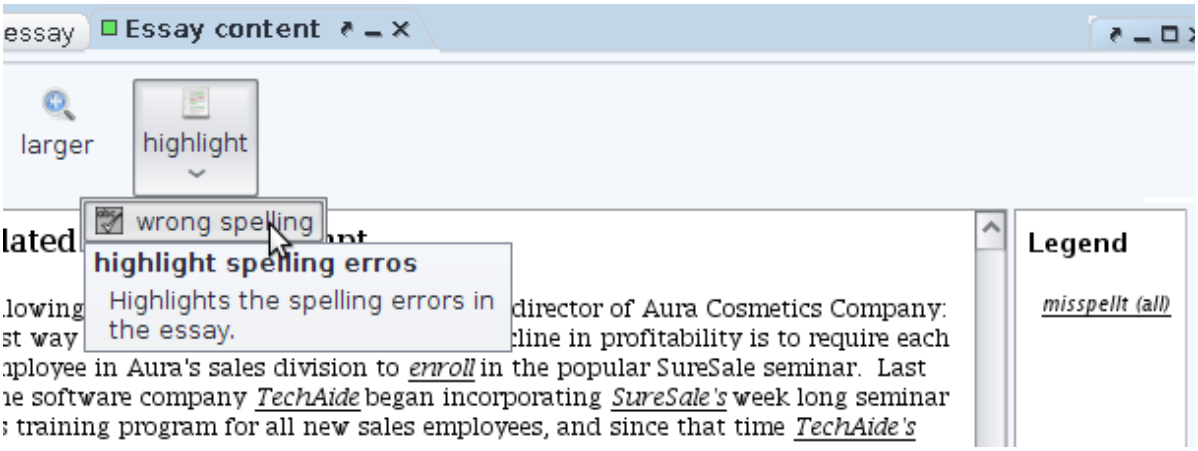


Figure 6.18.: Prototype Essay Feature Highlighting

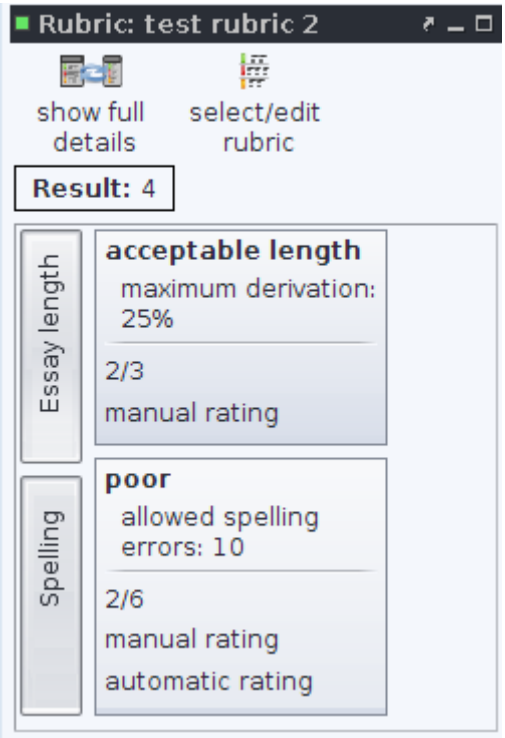


Figure 6.19.: Prototype Rated Rubric

Figure 6.19 shows the rubric rating for an essay. It is also clearly indicated in the condensed view which ratings are derived automatically and which have been overridden

by the user. Further details about the result may be provided by criteria as separate views. These are by default placed below the essay content view. When a criterion is clicked by the user a pop displaying all levels as seen in Figure 6.21 appears. This display can also contain an additional short explanation of the result for each criterion.

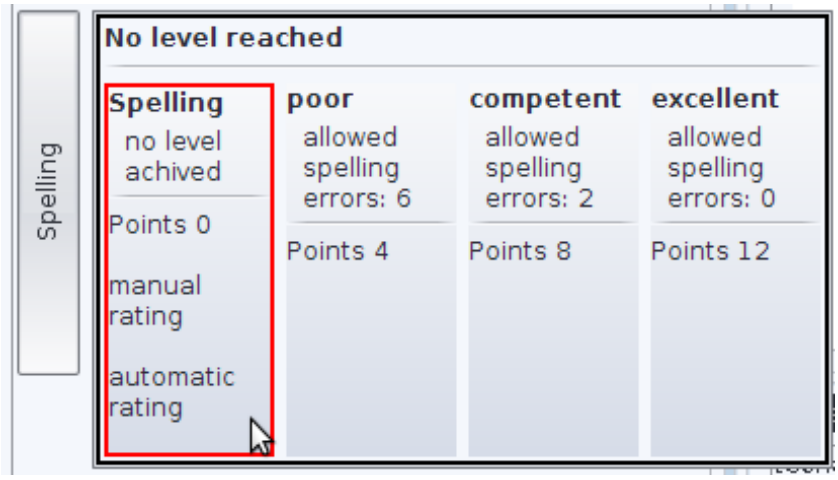


Figure 6.20.: Prototype Criterion Rating

Borders indicate automatic (dashed border) and manual ratings (solid border). In cases where an automatic rating is manually approved as seen in Figure 6.20 the border is solid to indicate that manual ratings override automatic ratings. It is not necessary to approve automatic ratings. This is implicitly assumed if no manual rating is provided for a given criterion.

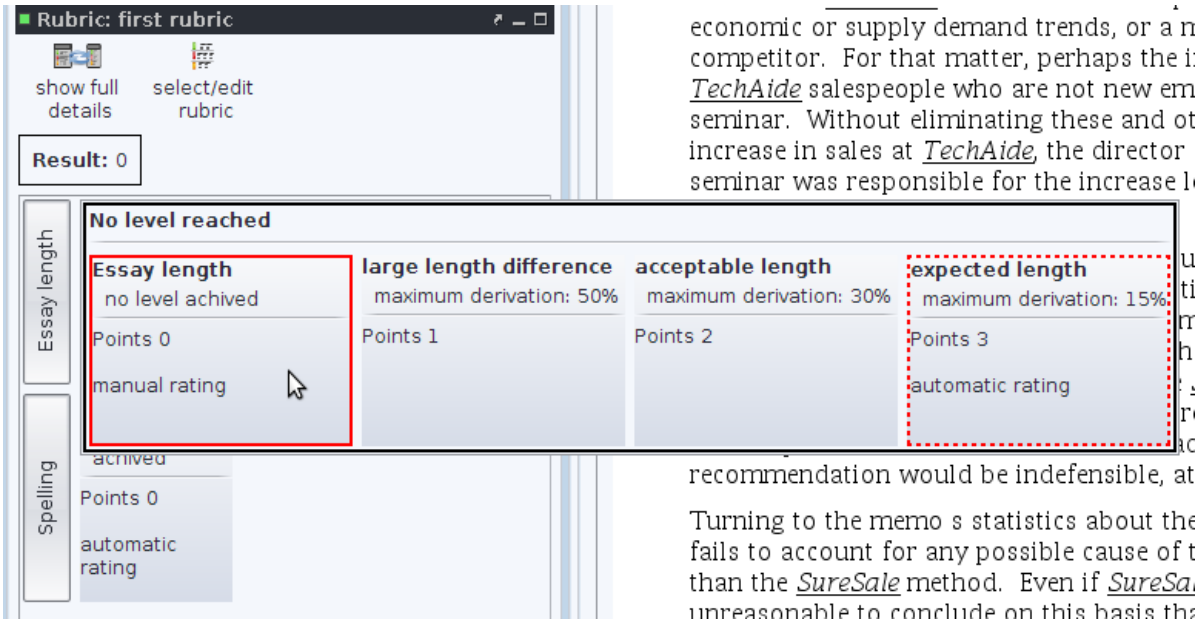


Figure 6.21.: Prototype Manual Criterion Rating

6.2. Evaluation

As outlined in Section 4.4.2 validity of the overall proposed system is dependent on the validity of the used criteria. So in the first part of the evaluation each implemented criterion is evaluated in the following sections. The second part of the evaluation is covering general system features as the semi-automatic grading approach and the GUI.

6.2.1. Spelling Criterion

The performance of a spell-checker is largely depending on the available dictionaries. In other usage scenarios like a word processing application the quality of correction suggestions is important, but for the usage as a rubric criterion the correct counting of wrong words is defining the desired quality. For validity this is still insufficient as false positives due to missing words in the used dictionary will be counted although they are correct.

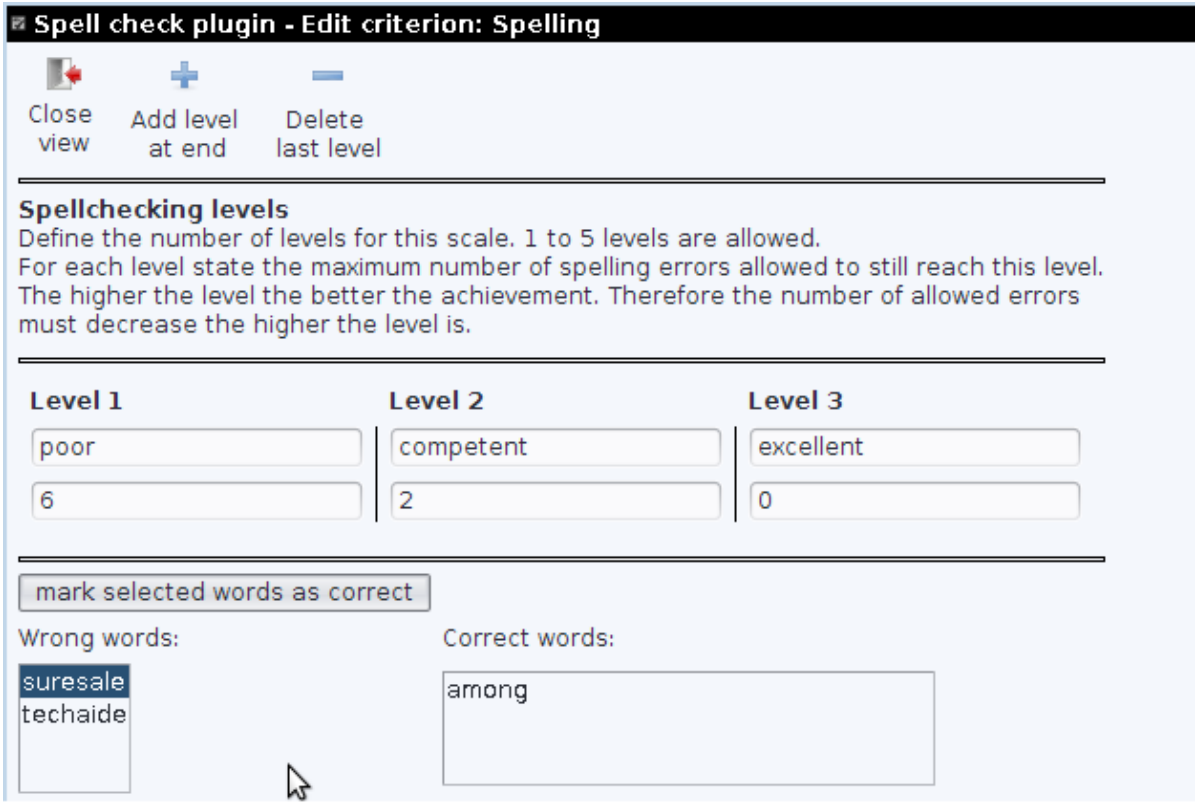


Figure 6.22.: Prototype Edit Spelling Criterion

This is counteracted in the implementation through the utilization of the example essays to discover domain specific vocabulary. A spell-check is performed on each of the positive example essays and all wrong words are added to a suggestion list as seen in Figure 6.22. The user can then easily select all the correct words and mark them as correct before the actual grading starts. This reduces the number of false positives

significantly resulting in fewer later corrections during the semi-automatic grading process. These marked words are not added to the dictionary but are stored as part of the configuration for the criterion for this specific rubric.

An open issue discovered through a user review is how wrong words are counted. The current implementation counts every wrong word. Some teachers might prefer to treat repeated word errors differently. An option specifying different counting strategies could be added to the criterion configuration.

6.2.2. Essay Length Criterion

The essay length criterion is one of the simplest but still tested to complete the evaluation. The example essays are used to calculable an average essay length which can be set as the standard for the evaluation. Alternatively a word count can be directly specified. Testing was done by selecting an essay as the ideal solution and then adding and removing portions of the text. The criterion was then applied to different text lengths delivering the expected results.

6.2.3. Readability Criterion

Evaluating the used readability formula itself is out of scope of this thesis. A literature review reveals positions for and against the usage of readability formulas in different applications (McGee, 2010; Dragan & Woo, 2010; Redish, 2000; Redish & Selzer, 1985). Teachers therefore should be informed about the limitations and correct usage of readability formulas before using them in essay evaluation.

Evaluating the criterion was done with a careful source code review to ensure the formula was correctly implemented. Secondly it was tested with there essays of which two were different types of text while the third was a fake nonsense text. Application of the criterion delivered the expected results appropriate for the classification of the texts.

6.2.4. Grammar Criterion

For the grammar criterion an external library was used to detect the errors. The library does not perform as well as commercial grammar checkers (Kies, 2010) but, as it is open source and uses a XML configuration format, rules can be adapted and new rules easily added. A check with various essays successfully detected grammar errors. As a special data set 20 test samples, mainly one sentence examples, were used. The data of the twenty most frequent errors comes from the original article published by R. J. Connors and Lunsford (1988), as cited in the grammar checker comparison by Kies (2010). The full data set can be found in Appendix G. In this test the LanguageTool library in version 1.0.0 found 9 out of the 20 problematic samples. Kies had reported a success rate of only 3 of 20 samples for LanguageTool version 0.9.1. The increase is a result of the active library development incorporating user provided improvements.

Teachers reading through essays finding more errors can easily override the basic automatic rating. As the automatic checking is lenient, the expected statistical outcome

of the manual overrides is that teachers find a higher number of grammar errors resulting in lower student scores for this criterion. Not enough essays and manual ratings could be gathered during the test runs to make a thorough statistical analysis to reassess this assumption.

6.2.5. Concept Criterion

To extract concept suggestions out of the example essays a web-service is used. For an evaluation of the extracted concepts refer to the original publication by Weinhofer (2010) or the application of the algorithm in the work by Gütl et al. (2011); Gütl, Lankmayr, and Weinhofer (2010).

The rubric concept criterion allows teachers to mark the found concepts in the text through highlighting (see Figure 6.23). All concepts can be highlighted at once or only the desired concept groups. As the evaluation algorithm does not only search for direct textual matches but utilizes WordNet to discover synonyms and related terms (for details see Section 5.5.11) individual highlighter actions for this matches are provided.

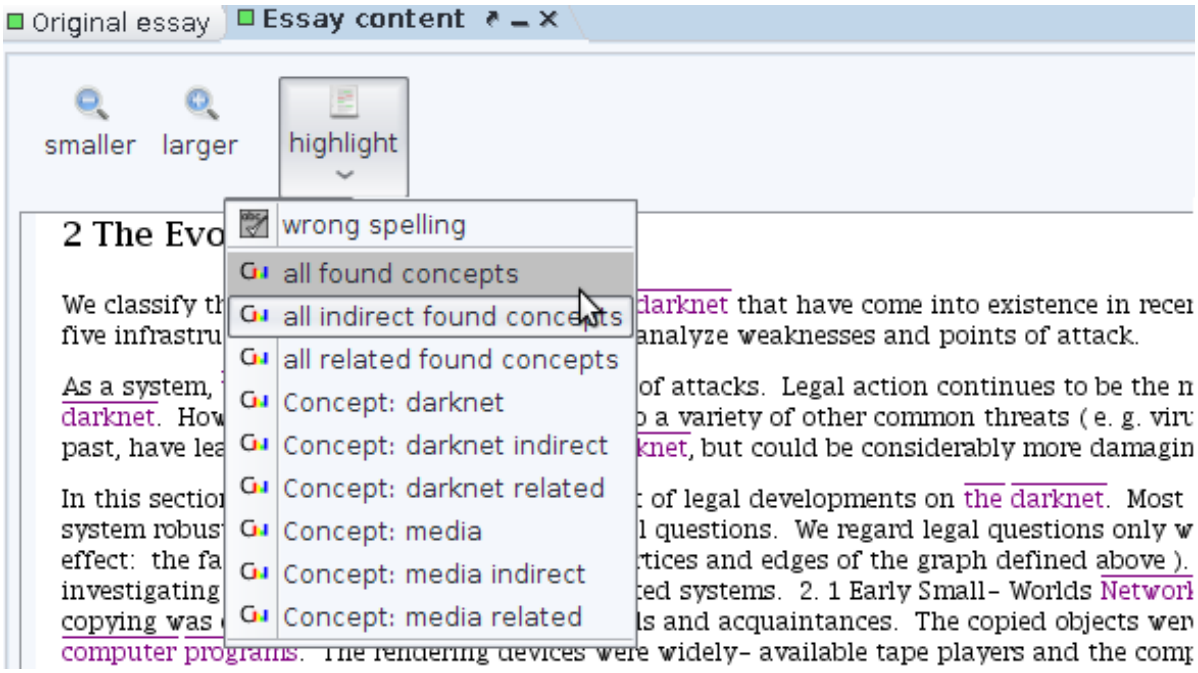


Figure 6.23.: Prototype Concept Criterion Highlighting

The concept discovery was tested with a set of texts retrieved through Wikipedia and Google on a specific topic. Although these texts would be better classified as articles rather than essays they still can be used to test the criterion. A document containing a listing of the specified concepts of the tested rubric was included as a control element. This document should receive a few points but never should receive the highest score. Due to the fact that the concept definitions in the rubric will never cover the whole content of an essay, this is the expected result. The criterion delivered the expected

results on the handmade test selection during the tests performed. As the web-service was only available very lately in the development cycle no extensive testing could be done. During the development only an example extract for one example essays was available which was used to simulate the web-service results. Therefore testing was done only with datasets related to this single example topic.

Another test with an essay using the same main concept name but covering a different concept was conducted. As the two different concepts not only shared the same name but both were related to computer networks this was an adequately difficult test-case. The expected result would be that the essay would gain a positive score but not be ranked highest. Results in this test case were mixed and heavenly depended on the concept definitions in the rubric. These results strongly suggest that the utilisation of the negative example could yield to better and more stable results. The current implementation only utilizes the positive example essays. Through the negative example essays off-topic concepts could be defined as negative concepts in the rubric definition. Matches on these concepts in essays evaluation then could lower the score a student essays receives.

6.2.6. Positive and negative Example Essays

Some of the currently implemented criteria successfully use the provided positive example essays. Examples for the practical application are the spelling and concept criterion. In spelling the usage can successfully reduce the number of false positives before the actual grading is done, greatly reducing the need to reevaluate essays if further correct words are found during essay reviews by the teachers. The essays are even more important in the concept criterion as they greatly ease the criterion configuration for teachers by providing concept suggestions derived from the example essays.

Limitations of the current prototype are that no criterion makes use of negative examples yet. Also the rubric cannot be easily applied to the example essays to test its validity. Currently the user needs to additionally import the example essays as student essays to be able to apply the rubric to them. This is a burden for users and support for the testing phase should be added to the system.

6.2.7. Semi-Automatic Evaluation

Semi-automatic grading is supported by the system as every automatic rating can be manually overridden. Besides the manual overrides teachers can utilize the possibility of automatic reevaluation. This is demonstrated best with the spelling criterion. Counting spelling errors is tiring for teachers and can be easily done by computers, but the problem of false positives through domain specific vocabulary not contained in the dictionary limits the preciseness of automatic results. As outlined in Section 6.2.1 false positives can be successfully reduced through example essays beforehand.

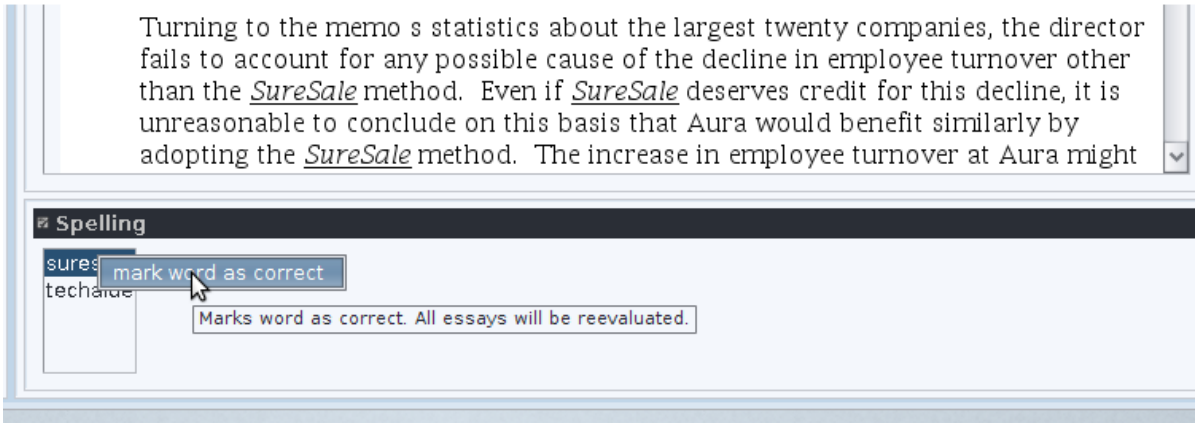


Figure 6.24.: Prototype Spelling Reevaluation

Students can still use further correct vocabulary missing in the dictionary. Figure 6.24 shows the detail pane provided through the spelling criterion for the currently displayed essay. Teachers can mark words as correct there and the automatic result will be updated immediately. All other essays which have previously been automatically evaluated are marked for reevaluation. As soon as the teachers views one of them the automatic result will be recalculated for the internally marked criteria needing reevaluation. This is faster compared to initial automatic rating as only necessary calculations are redone. Manual results are not affected by the automatic reevaluation and are kept as they are.

When testing the prototype users reported the need to clearly indicate which essays had been already manually reviewed and which not. Users are assisted by the system as by default the last state is restored which includes the last selected essay. If teachers would linearly advance through the essays, it is easy to ensure that all essays have been reviewed. As this might not be the case a solution to hint teachers to the not reviewed essays is missing in the prototype which must be part of a final system.

6.2.8. GUI

During the prototype development, aside from the own heuristic evaluations done (see Section 5.5.6), an additional evaluation using recognized usability principles was done by an experienced software developer not involved in prototype development. As the evaluation was done in about the middle of the development process, all found issues could be solved in the further prototype development.

Close to the finishing of the prototype development another GUI evaluation round was started. For this review two users have been selected to familiarize themselves with the system and perform a set of tasks. The used method were think aloud protocols (Lewis & Rieman, 1994). The two think aloud GUI evaluation sessions have been typed live. Later they were checked against an audio recording[1] and have been supplemented were necessary. The protocols can be found in Appendix H.

[1]Users were assured that the audio recording would be solely used to check the notes taken and after that be deleted.

User 1 was the same experienced software developer conducting a previous test in the middle of development. The user received no further instructions or information about the system than what was remembered from the previous session months ago. User 2, a former mathematics high-school teacher, had not seen the system before. The user was also not familiar with the concept of using rubrics for grading as rubrics are less common in the Austrian school system. Therefore a short introduction about the goal and purpose of the software was given to the user, along with instructions on the think aloud testing procedure. Both users were asked to summarize their impressions at the end.

Through the final GUI evaluation some issues were found. Some of them still could be fixed in the prototype system while a few remain to be solved in future development. The following paragraphs present the findings derived from the preformed user GUI tests.

Different backgrounds of users lead to different user interface experiences as could be observed in two striking examples. User 2 expressed the requirement to be able to differentiate (weight) the distinct criteria already in the pre-instructions given. Therefore the weight configuration for the criteria was naturally used by the user during the task to prepare a rubric for grading. User 1 lacked the assessment background teachers have and therefore did not immediately understand the function of weighting the criteria.

The second example is based on the background of user 1 as a software developer. The user had a good understanding of the concept of plugins and therefore performed the activation and usage of the available plugins without any delay. The user was able to add the newly available criteria to the rubric without workflow disruption. User 2 experienced major difficulties in this part of rubric creation. Although the plugins were successfully activated as suggested during the wizard guiding the rubric creation, the user failed to add the wanted criteria to the rubric being constructed.

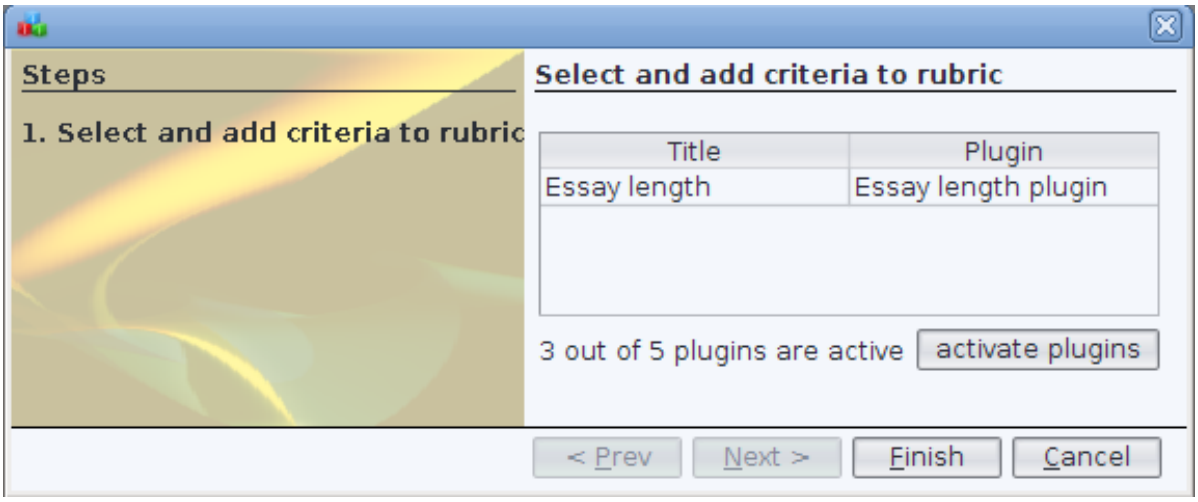


Figure 6.25.: Prototype GUI Test Add Criterion

The problem lies in the wizard dialog as seen in Figure 6.25. The list contains the currently available criteria which have not been used yet in the current rubric. The

criteria list is based on the activated plugins, so criteria by currently deactivated plugins are not shown. User 2 thought that through the successful plugin activation the criterion was already added to the rubric. But the shown dialog requires the user to select the wanted criterion in the list to be added to the rubric. The dialog needs to be redesigned so that users clearly see what criteria are used in the rubric and which can still be added to the rubric.

Similarly the list to select a rubric for the current assignment should be redesigned. Both users first lacked the information that it is possible to define several rubrics for one assignment. This requires to select the rubric to be used leading to an explicit selection step to use a rubric. A simple first solution would be that if only one rubric is available it is automatically selected.

User 1 expressed concerns about the dialog to find a specific student submission. The user found it unclear which data is searchable to find a specific submission. The dialog used in the system is the default dialog provided by the used SWING table component. It either should be replaced or extended to show which information units are searchable.

User 2 expressed the wish to be able to mark parts of the text and associate them with a criterion. In the case of spell-checking the user expected to be able to correct the automatic result by easily providing input to the automatic evaluation this way. He also expressed the wish to be able to add a comment for the student at the same time. Again this results from the background as a teachers whereas user 1 did not express similar intentions.

Both users agreed that they would be confident to use the overall system after they did one short training session with a few instructions by a person familiar with the system. The only concern raised was the configuration of the different criteria algorithms. Users expressed some insecurity for some configuration like for example in the concept criterion. Both users felt the help information provided by tooltips very helpful and sufficient. Only the provided help for configuring the criteria was, as in the example of concept criterion, sometimes perceived as insufficient and therefore needs more expressive explanations.

Another point needing further documentation or help hints is the IDE style window layout. Both user saw the possibility to save and restore layouts but rather ignored it in the first place. User 2 did not utilize the possibility to resize and move parts of the GUI at all. User 1 immediately used it after the expressed dissatisfaction that the content of one view was not completely visible. User 2 did not experience this problem due to a larger screen available at the used test system. Besides the different screen sizes in the test setup, this is again partly based on the different user backgrounds were user 1 knows the utilized IDE views concept from daily work. Concluding this seems to be the only part of the main interface which needs better documentation for inexperienced users.

6.3. Summary

The usage instruction clearly demonstrated the intended usage scenario for the developed system. The GUI user testing in Section 6.2.8 showed that the system can be efficiently used after only one training session. Still for a real system there are missing features to

effectively support teachers and achieve a higher system acceptance.

The prototype implementation has several general system limitations which prevent actual real life usage not considering the criteria implementations themselves. These are mainly limitation in the GUI. For example the current prototype does not restrict the change of the used rubric until the result publishing deadline is reached. In a real usage scenario a rubric must be fixed and cannot be changed while actual student essays are graded. Ideally after the rubric is defined and tested it should be marked as fixed and might be published to the students before the actual grading is done.

A limitation of the prototype is the lack to support rubric testing while defining a new rubric. Currently teachers need to add test essays as they would add student essays and then apply the rubric to these test essays. After reviewing the results, they can edit the rubric and repeat the testing. Finally they need to remove the test essays from the essay list before actual student essays are imported. The testing phase could be much better supported by the system by providing a test mode where test essays are stored in a separate list avoiding the need to remove them after testing. Configuring the criteria could then be partly done automatic if the teachers manually grade the test essays and the criteria implementation use the provided data to fine tune their internal configuration especially focusing on the level specifications.

The missing LMS integration is a major obstacle for real life application as essay importing is rather tedious in the prototype. Batch processing could be added but is considered a low priority as an integration into existing systems managing student submissions, deadlines and grades is of much greater overall value.

As outlined in Section 6.2.7 semi-automatic grading is well supported but some convenience features for users as hinting which essays have not been manually reviewed yet are missing. Also the GUI user testing in Section 6.2.8 showed that the developed prototype is well suited for the semi-automatic grading approach. Some possible obstacles for users have been found in the GUI which can be easily solved in further development.

The implemented criteria have been evaluated and found to deliver the expected results. What is still missing is a test using a larger test data set and several users grading all essays. Such a test will provide further data to enhance the implemented criteria and also will deliver more input for GUI optimisations. For most of the criteria possible directions for further improvements have been already found in addition to the ones described in Section 5.5.

7. Lessons Learned

This chapter describes lessons learned during the thesis project and is structured into three parts covering the literature research, prototype implementation and usage.

The literature review showed the importance of assessment in the learning process. The importance varies in the different learning theories and is especially high in formative practises found in modern learning settings. Assessment can be done in a variety of forms and we learned, that instructional design needs to consider which assessment types are suitable for a certain task, as different forms may cover distinct ranges of assessed aspects. My experience in computerized assessment was limited to the application in programming classes before this project. The literature review greatly broadened my knowledge about this area and existing solutions. It also showed the limitations of the existing approaches, which are therefore mainly used in specific settings. Another new input was the usage of assessment rubrics which is far more common in other countries compared to Austria. Although their application is fairly easy we learned that the construction of a rubric needs careful design to deliver valid results. Therefore practise in rubric design as well as trial applications of a newly designed rubric are helpful to achieve reliable outcomes.

During the implementation phase we valued the support of modern IDEs to refactor code. This is extremely helpful when a project cannot be fully specified in every detail from the beginning, as it is common in research projects. The major lesson learned was how hard proper GUI design can be. On first glance it seems quite easy and there are good guidelines one can follow during the development. Even when these guidelines are strictly followed the outcome often will not be satisfactory, especially in situations where uncommon tasks need to be solved. Therefore different GUI evaluation techniques were reviewed. Usability was seen as a highly important requirement for the project, but as this was not the main topic of the thesis, two review techniques which could be easily integrated into the development process had been used. These proofed to be very helpful in discovering insufficient solutions in the user interface and better approaches could be implemented in most cases.

Another lesson we learned is that the time spent choosing the tools to work with pays off multiple times. To hurry with evaluating and deciding in tool and library selection can cost you much more time later. While the balance here is easier to find for tools like the IDE it is much harder in case of code libraries. The problem there is that looking at sample code, the documentation and doing a short trial is often not enough basis for a sound decision, although that takes already considerable time. The big issues, like surprising limitations, often only show up during the real development. This happened a few times during code development and has cost quite some time despite the fact, that design patterns which mitigate these problems were used. Though we knew this

principally through the previous experience during my studies, it still had a lasting impression on me. It seems wise to me now to even spend one day more in actually trying out the libraries, going into more depth in using the code, prior to actually integrating them into the project.

The development process also showed the need to set up testing environments as quickly and easily as possible. First the prototype only used a MYSQL database for storage but this proved to be an obstacle for speedily deploying test environments. Therefore a locally integrated database solution was searched and found in Apache Derby which can also be accessed with SQL. This speeded up the testing greatly but another drawback connected to such an internal solution was revealed. In the case of an external database it is possible to access the database with available standard tools during the development to check the stored data. For the internal Derby database this was not easily possible. Therefore for debugging purposes a simple tool was developed to access the internal database of a deployed prototype set up. This was a personal experience how many helpful tools have been started as a side project out of needs showing up during the development of the main project.

GUI evaluation was also done in the last phase as part of the overall prototype evaluation. It seems that one training session is enough for previously inexperienced users to be able to use the system confidently aside from the criteria configuration for automatic evaluation. This and as the system is applicable at the classroom level are indications that a proper design can successfully deliver the planned outcomes. The project therefore stressed my personal believe that careful system design is necessary in the software development process, but that one must consider that the initial design might change during the development, especially in a research context.

8. Summary and Outlook

The aim of the project was to develop an essay assessment system which can be used at the classroom level in formative learning settings. This idea is based on the insights gained through the literature review done in Chapter 2. Assessment is a vital part of the learning process and therefore needs careful consideration in teaching. The various assessment types fit different goals in assessment and therefore an appropriate selection has to be done in instruction design. Essays are a valid choice in a broad variety of subjects and can be used to test high-order skills. Besides the question type the mode and tools used for assessment influence the outcomes as well. Analytical rubrics were found to be an appropriate tool for subjective type questions as essays. Rubrics have the advantage that they explicitly state the required performance goals for students and therefore contain valuable feedback.

As assessment is considerable effort for teachers, supporting techniques are of interest. Therefore existing systems in the context of essay assessment have been reviewed in Chapter 3. It was found that most solutions are applicable only when a high number of essays need to be graded, lacking the support at the classroom level. Based on these outcomes a proposed solution usable at the classroom level and for formative assessment was introduced in Chaper 4. Chapter 5 covered the details of the developed prototype. The usage of the prototype is documented in Chapter 6 along with an evaluation of the system.

The semi-automatic solution developed is usable at the classroom level with lower student numbers and can successfully reduce the time spent on grading assignments. This makes it possible to perform more assessments enabling teachers to adapt formative practises in their courses. The usage of an analytical rubric as the assessment base makes the whole assessment process more transparent. The developed prototype allows teachers to grade essays without using the automatic features at least as fast as on paper. As support through automatic analysis is available teachers are positively encouraged to utilize the system and the time needed for grading is further reduced.

Future Work To reach a fully usable system the developed criteria need to be enhanced further to deliver even conciser results. Additionally more criteria need to be implemented to make the system applicable to a greater range of essays. Besides the improvements in essay analysis the system needs to be integrated into a LMS for efficient trial runs as manually importing student submissions encumbers a more thorough evaluation of the overall system performance.

Aside from the direct system improvements further chances exists in the broader application of the system. Assessment done by teachers is not the only assessment option

used nowadays. Different assessment models exist and some foster student self assessment and peer assessment in modern learning settings which can be efficiently used in e-assessment (AL-Smadi, Gütl, & Chang, 2011; AL-Smadi, Gütl, & Kappe, 2010). The proposed system could be adapted to be used in such a self or peer-assessment mode as well. This would be best done as a service approach as described by the Service-Oriented Framework for Assessment (SOFA) (AL-Smadi, Gütl, & Helic, 2009). This would allow the system to be used in a greater variety of different applications beyond the planned basic LMS integration.

One such greatly differing application would be the use in 3D virtual learning environments. In such a scenario the results of an assessment should be available in the environment the students already use and therefore be presented in world as well. An SOA approach enables that the grading part can remain outside the virtual learning environment. A similar idea is found in a prototype system reported by (Gütl, Chang, & Freudenthaler, 2010) in wich a web interface is used to manage and manipulate learning settings and artefacts which then are displayed in the Second Life 3D world platform. Similarly an integration could manage a rubric artefact in the world to communicate the results of an assessment to students logged into the world.

References

ACT. (2009). *Act national curriculum survey® 2009.* http://www.act.org/research/policymakers/pdf/NationalCurriculumSurvey2009.pdf.

Ala-Mutka, K., & Järvinen, H.-M. (2004). Assessment process for programming assignments. In Kinshuk et al. (Eds.), *Icalt* (p. 181 - 185). Washington, DC: IEEE Computer Society.

Allen, D., & Tanner, K. (2006). Rubrics: Tools for making learning goals and evaluation criteria explicit for both teachers and learners. *CBE Life Sciences Education*, *5*(3), 197-203.

AL-Smad, M., & Gütl, C. (2008). Past, present and future of e-assessment: Towards a flexible e-assessment system. In M. Auer (Ed.), *Cd-rom.* Kassel 34127, Germany: Kassel University Press. Available from http://www.iicm.tugraz.at/home/cguetl/publications/2008/Al-Smadieetal.2008-CAF-ICL.pdf (http://www.upress.uni-kassel.de/publi/abstract.php?978-3-89958-353-3)

AL-Smadi, M., Gutl, C., & Kannan, R. (2010, February). Modular assessment system for modern learning settings: Mass. *International Journal of Computer Applications*, *1*(9), 43-49. Available from http://www.ijcaonline.org/journal/number9/pxc387342.pdf (Published By Foundation of Computer Science)

AL-Smadi, M., Gütl, C., & Chang, V. (2011, March). Addressing e-assessment practices in modern learning settings: A review. In *Global learn asia pacific 2011.* Melbourne, Australia.

AL-Smadi, M., Gütl, C., & Helic, D. (2009). Towards a standardized e-assessment system: Motivations, challenges and first findings. *International Journal of Emerging Technologies in Learning (iJET)*, *4*(Special Issue: IMCL2009), 6-12.

AL-Smadi, M., Gütl, C., & Kappe, F. (2010, july). Towards an enhanced approach for peer-assessment activities. In *Advanced learning technologies (icalt), 2010 ieee 10th international conference on* (p. 637 -641).

Andrade, H. G. (2000, February). Using rubrics to promote thinking and learning. *Educational Leadership*, *57*(5), 13-18. Available from http://www.ascd.org/publications/educational_leadership/feb00/vol57/num05/Using_Rubrics_to_Promote_Thinking_and_Learning.aspx

Andrade, H. G., & Du, Y. (2005). Student perspectives on rubric-referenced assessment. *Practical Assessment, Research & Evaluation*, *10*(3). Available from http://pareonline.net/getvn.asp?v=10&n=3

Anglin, L., Anglin, K., Schumann, P. L., & Kaliski, J. A. (2008, February). Improving the efficiency and effectiveness of grading through the use of computer-assisted grading rubrics. *Decision Sciences Journal of Innovative Education*, *6*(1), 51-73.

Arter, J. A., & McTighe, J. (2001). *Scoring rubrics in the classroom: Using performance criteria for assessing and improving student performance* (T. R. Guskey & R. J. Marzano, Eds.). Thousand Oaks, Calirfornia 91320: Corwin Press.

Atkin, J. M., Black, P., & Coffey, J. (Eds.). (2001). *Classroom assessment and the national science education standards.* Washington, DC: National Academy Press.

(Committee on Classroom Assessment and the National Science Education Standards)

Attali, Y., & Burstein, J. (2006, February). Automated essay scoring with e-rater® v.2. *Journal of Technology, Learning and Assessment*, *4*(3). Available from http://escholarship.bc.edu/jtla/vol4/3/

Austin, M. J., & Brown, L. D. (1999). Internet plagiarism: Developing strategies to curb student academic dishonesty. *The Internet and Higher Education*, *2*(1), 21 - 33. Available from http://www.sciencedirect.com/science/article/B6W4X-3YPC6PR-4/2/0221af33f8b904ae9bbce07d49966536

A.V. v. iParadigms, LLC. (2009, April). Retrieved 30.12.2010 from http://pacer.ca4.uscourts.gov/opinion.pdf/081424.P.pdf. (United States Court of Appeals for the Fourth Circuit)

Barrett, S. (2001). The impact of training on rater variability. *International Education Journal*, *2*(1), 49-58. Available from http://ehlt.flinders.edu.au/education/iej/articles/v2n1/barrett/barrett.pdf

Ben-Simon, A., & Bennett, R. E. (2007, August). Toward more substantively meaningful automated essay scoring. *Journal of Technology, Learning, and Assessment*, *6*(1). Available from http://escholarship.bc.edu/jtla/vol6/1/

Bereiter, C. (2003). Foreword. In M. D. Shermis & J. C. Burstein (Eds.), (p. VII-X). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Birenbaum, K., M.and Breuer, Cascallar, E., Dochy, F., Dori, Y., Ridgway, J., Wiesemes, R., et al. (2006). A learning integrated assessment system. *Educational Research Review*, *1*(1), 61-67.

Black, P., & Wiliam, D. (1998, October). Inside the black box: Raising standards through classroom assessment. *Phi Delta Kappan*, *80*(2), 139-148. Available from http://web.archive.org/web/20060404074132/http://pdkintl.org/kappan/kbla9810.htm

Black, P. E. (2007). metaphone. In *Dictionary of algorithms and data structures (online).* Gaithersburg, MD: U.S. National Institute of Standards and Technology. Retrieved 20.11.2012 online from http://xw2k.nist.gov/dads/html/metaphone.html.

Blake Dawson Waldron. (2004, April). *Turnitin australian legal document.* Retrieved 30.12.2010 from http://turnitin.com/static/pdf/australian_legal.pdf.

Blood, E., & Spratt, K. F. (2007). Disagreement on agreement: Two alternative agreement coefficients. In *Proceedings of the sas global forum 2007 conference.* Cary, NC: SAS Institute Inc. Available from http://www2.sas.com/proceedings/forum2007/186-2007.pdf

Bloom, B. S. (Ed.). (1956). *Taxonomy of educational objectives, book 1: Cognitive domain.* White Plains, N.Y.: Longman.

Blue Wren. (n.d.). *Markit online demo (my account).* Retrieved 11.3.2010 from http://www.essaygrading.com/registered/jobList.faces.

Bransford, J. D., & Brown, R. R., Ann L.and Cocking (Eds.). (2000). *How people learn: Brain, mind, experience, and school* (expanded ed.). Washington, DC: National Academy Press. Paperback. (Committee on Developments in the Science of Learning with additional material from the Committee on Learning Research &

Educational Practice)

Bresciani, M. J., Oakleaf, M., Kolkhorst, F., Nebeker, C., Barlow, J., Duncan, K., et al. (2009, May). Examining design and inter-rater reliability of a rubric measuring research quality across multiple disciplines. *Practical Assessment, Research & Evaluation*, *14*(12). Available from http://pareonline.net/getvn.asp?v=14&n=12

Brookhart, S. M. (1999). *The art and science of classroom assessment: The missing part of pedagogy* (Vol. 27) (No. 1). Washington: George Washington University.

Brualdi, A. (1998). Implementing performance assessment in the classroom. *Practical Assessment, Research & Evaluation*, *6*(2). Available from http://pareonline.net/getvn.asp?v=6&n=2

Bruer, J. T. (1994). *Schools for thought: A science of learning in the classroom.* The MIT Press, Cambridge, Massachusetts 02142: The MIT Press.

Bull, J., & McKenna, C. (2004). *Blueprint for computer-assisted assessment.* New York: RoutledgeFalmer.

Burstein, J. (2003). The e-rater® scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis & J. C. Burstein (Eds.), (p. 107-117). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., Braden-Harder, L., et al. (1998). Automated scoring using a hybrid feature identification technique. In *Proceedings of the 36th annual meeting of the association for computational linguistics and 17th international conference on computational linguistics* (Vol. 1, p. 206-210). Montreal, Canada: Association of Computational Linguistics. Available from http://www.aclweb.org/anthology-new/P/P98/P98-1032.pdf

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern - oriented software architecture - a system of patterns.* Chichester, West Sussex, England: John Wiley & Sons Ltd.

Butakov, S., & Scherbinin, V. (2009, May). The toolbox for local and global plagiarism detection. *Computers and Education*, *52*(4), 781-788.

Carter, J., Ala-Mutka, K., Fuller, U., Dick, M., English, J., Fone, W., et al. (2003). How shall we assess this? In *Iticse-wgr '03: Working group reports from iticse on innovation and technology in computer science education* (p. 107-123). New York, NY, USA: ACM.

CBC News. (2004, Jan. 16). *Mcgill student wins fight over anti-cheating website.* Retrieved 31.12.2010 from http://www.cbc.ca/news/story/2004/01/16/mcgill_turnitin030116.html.

Chang, V., & Gütl, C. (2010, May). Generation y learning in the 21st century: Integration of virtual worlds and cloud computing services. In Z. W. Abas, I. Jung, & J. Luca (Eds.), *Proceedings of global learn asia pacific 2010* (p. 1888-1897). Penang, Malaysia: AACE.

Chen, C.-F. E., & Cheng, W.-Y. E. (2008, June). Beyond the design of automated writing evaluation: pedagogical practices and perceived learning effectiveness in efl writing classes. *Language Learning & Technology*, *12*(2), 94-112. Available from http://llt.msu.edu/vol12num2/chencheng.pdf

Chung, G. K. W. K., & O'Neil, H. F. J. (1997). *Methodological approaches to*

online scoring of essays (Tech. Rep.). Los Angeles, CA: National Center for Research on Evaluation, Standards, and Student Testing, Los Angeles, CA. Available from http://www.eric.ed.gov/ERICWebPortal/contentdelivery/servlet/ERICServlet?accno=ED418101 (National Center for Research on Evaluation, Standards, and Student Testing, Los Angeles, CA)

Churchill, L. (2005, Dec.). Students: 2, turnitin: 0 - senate committee rules in favour of student two years after she failed class for refusing to submit paper to turnitin.com. *The McGill Daily*, *95*(25). Retrieved 31.12.2010 from archive.org http://web.archive.org/web/20070517093213/http://www.mcgilldaily.com/view.php?aid=4615, orignally published at http://www.mcgilldaily.com/view.php?aid=4615.

Connors, P. (2008, May). Assessing written evidence of critical thinking using an analytic rubric. *Journal of Nutrition Education and Behavior*, *40*(3), 193-194.

Connors, R. J., & Lunsford, A. A. (1988). Frequency of formal errors in current college writing, or ma and pa kettle do research. *College Composition and Communication*, *39*(4), 395-409.

CTB/McGraw-Hill LLC. (n.d.-a). *Writing roadmap - ctb/mcgraw-hill: Onlince student training demo.* Retrieved 15.4.2011 from, http://www2.ctb.com/online_demo_wrm2/wr_student_trainer/pages/return.htm.

CTB/McGraw-Hill LLC. (n.d.-b). *Writing roadmap - ctb/mcgraw-hill: Onlince teacher training demo.* Retrieved 15.4.2011 from, http://www2.ctb.com/online_demo_wrm2/wr_teacher_trainer/pages/mainmenu.htm.

Dewey, J. (1997). *Experience and education* (First ed.). 1230 Avenue of the Americas, New York, NY 10020: Touchstone. (Originally published by Kappa Delta Pi 1928)

Dochy, F., Gijbels, D., & Segers, M. (2006). Instructional psychology: Past, present and future trends. In L. Verschaffel, F. Dochy, M. Boekaerts, & S. Vosniadou (Eds.), (p. 191-206). Oxford: Elsevier.

Dragan, M., & Woo, A. (2010, Feb.). *The methodology used to assess the readability of the NNAAP examination* (Tech. Rep.). Chicago, IL: National Council of State Boards of Nursing. Available from https://www.ncsbn.org/Readability_of_NNAAP.pdf

Dreher, H. (2007). Automatic conceptual analysis for plagiarism detection. *Journal of Issues in Informing Science and Information Technology*, *4*, 601-614.

DuBay, W. H. (2004, August). *The principles of readability.* Retrieved 4.12.2010 from http://www.impact-information.com/impactinfo/readability02.pdf. (Impact Information, Costa Mesa, California)

Earl, D. (2007, Nov.). Concepts. In J. Fieser, B. Dowden, & R. Gennaro (Eds.), *Internet encyclopedia of philosophy.* Martin, U.S.A: University of Tennessee. (Retrieved 27.12.2010 from http://www.iep.utm.edu/concepts/)

Earl, L. M. (2003). *Assessment as learning: Using classroom assessment to maximize student learning* (T. R. Guskey & R. J. Marzano, Eds.). Thousand Oaks, California: Corwin Press.

Educational Testing Service. (n.d.). *Criterion online tour.* Retrieved 8.2.2010 from http://www.ets.org/Media/Products/Criterion/tour2/critloader.html.

Elliot, S. (2003). Intellimetric™: From here to validity. In M. D. Shermis & J. C. Burstein (Eds.), (p. 67-81). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Exam question types and student competencies. (n.d.). Retrieved 26.7.2010 from `http://teachingacademy.wisc.edu/archive/Assistance/course/essay.htm`. Madison, WI.

Family educational rights and privacy act (ferpa). (2004, July). 20 U.S.C. § 1232g. Available from `http://www.access.gpo.gov/nara/cfr/waisidx_04/34cfr99_04.html` (codified at C.F.R 34, part 99)

Flesch, R. (1979). *How to write plain english: A book for lawyers and consumers.* New York: Harper and Row. (Excerpt Chapter 2 retrieved 13.12.2010 from `http://www.mang.canterbury.ac.nz/writing_guide/writing/flesch.shtml`)

Fl statutes - title xxxvii insurance section 627.011. (2004, July). Available from `http://law.onecle.com/florida/insurance/627.4145.html` (Florida Laws)

Forehand, M. (2005). *Bloom's taxonomy.* Retrieved 20.7.2010 from `http://projects.coe.uga.edu/epltt`. Bloomington, IN: Association for Educational Communications and Technology. Available from `http://www.aect.org/Intranet/Publications/index.asp`

Foster, A. L. (2002, May 17). Plagiarism-detection tool creates legal quandary. *The chronicle of higher education*. Retrieved 30.12.2010 from `http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/CHRON_HE/C020517F.pdf`. Available from `http://chronicle.com/free/v48/i36/36a03701.htm`

Freeman, R., & Lewis, R. (1998). *Planning and implementing assessment.* New York: RoutledgeFalmer.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. M. (1995). *Design patterns: Elements of reusable object-oriented software* (28th printing 2004 ed.). Indianapolis, IN: Addison-Wesley.

Gateway Software Productions. (2006). *The rubric builder.* Retrieved 8.2.2010 from `http://www.rubricbuilder.com/app/viewedit.aspx`.

Gijbels, D., & Dochy, F. (2006, Dec.). Students' assessment preferences and approaches to learning: can formative assessment make a difference? *Educational Studies*, *32*(4), 399-409.

Gmat® scores and score reports. (n.d.). Retrieved 10.2.2010 from, `http://www.mba.com/mba/thegmat/gmatscoresandscorereports`. (Graduate Management Admission Council)

Golding, A. R., & Roth, D. (1999). A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, *34*(1), 107-130.

Grademark paperless grading improvements. (2010, Sep.). Retrieved 30.12.2010 from, `http://turnitin.com/static/whatsnew/gradeMark.php`. (iParadigms, LLC)

Gruner, S., & Naven, S. (2005). Tool support for plagiarism detection in text documents. In *Proceedings of the 2005 acm symposium on applied computing* (p. 776-781). New York, NY, USA: ACM.

Gwet, K. (2001). *Handbook of inter-rater reliability: How to measure the level of agreement between 2 or multiple raters.* New York: STATAXIS Publishing Company.

Gütl, C. (2008a, February). Automatic limited-choice and completion test creation, assessment and feedback in modern learning processes. In *Memorias 2a. conferencia internacional de e-learning integral y 6ta. conferencia de openacs y .lrn.* Guatemala C.A..: Universidad Galileo. Available from http://ges.galileo.edu/fs/download/MemoriasConferenciaUniversidadGalileo?file_id=910694

Gütl, C. (2008b). Moving towards a fully automatic knowledge assessment tool. *International Journal of Emerging Technologies in Learning*, *3*(1). Available from http://online-journals.org/i-jet/article/view/172/240

Gütl, C., Chang, V., & Freudenthaler, S. (2010, Sep.). How to support more flexible learning settings in second life. In *Proceedings of the icl 2010* (p. 129-141). Hasselt, Belgium.

Gütl, C., Lankmayr, K., & Weinhofer, J. (2010, Nov.). Enhanced approach of automatic creation of test items to foster modern learning setting. In P. Escudeiro (Ed.), *Proceedings of the 9th european conference on e-learning* (Vol. 1, p. 225-234). Porto, Portugal: Academic Publishing Limited, Reading, UK.

Gütl, C., Lankmayr, K., Weinhofer, J., & Hofler, M. (2011, April). Enhanced approach of automatic creation of test items to foster modern learning setting. *Electronic Journal of e-Learning*, *9*(1), 23-38. Available from http://www.ejel.org/issue/download.html?idArticle=165 (Project ALICE "Adaptive Learning via Intuitive/Interactive, Collaborative and Emotional System")

Gütl, C., Pivec, M., Trummer, C., García-Barrios, V. M., Mödritscher, F., Pripfl, J., et al. (2005). Adele (adaptive e-learning with eye-tracking): Theoretical background, system architecture and application scenarios. *European Journal of Open, Distance and E-Learning (EURODL)*, *II*. Available from http://www.eurodl.org/?p=archives&year=2005&halfyear=2&article=197 (AdeLE project, http://adele.fh-joanneum.at)

Jackson, D. (2000). A semi-automated approach to online assessment. *SIGCSE Bull.*, *32*(3), 164-167.

James, C. L. (2006). Validating a computerized scoring system for assessing writing and placing students in composition courses. *Assessing Writing*, *11*(3), 167-178.

James, R., McInnis, C., & Devlin, M. (2002). *Assessing learning in australian universities*. Melbourne, Victoria 3010, Australia: Centre for the Study of Higher Education. Available from http://www.cshe.unimelb.edu.au/assessinglearning/docs/AssessingLearning.pdf

Johnson, M., & Greatorex, J. (2008). Judging text presented on screen: implications for validity. *E-Learning and Digital Media*, *5*(1), 40-50.

Johnson, M., Nádas, R., Bell, J. F., & Green, S. (2009, Sep.). Marking essays on screen: an investigation into the reliability of marking extended subjective texts. In *Iaea conference, 13 – 18 september 2009, brisbane.* Brisbane, Australia. Available from http://www.rmassessment.co.uk/sites/default/files/Cambridge%20Assessment%20research%20paper%20IAEA%20Conference%202009_1.pdf (Also published in The British Journal of Educational Technology' (2009))

Jonsson, A., & Svingby, G. (2007, May). The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review*, *2*(2), 130-144.

Kakkonen, T., Myller, N., & Sutinen, E. (2004). Semi-automatic evaluation features in computer-assisted essay assessment. In V. Uskov (Ed.), *Cate* (p. 456-461). Calgary, Canada: ACTA Press.

Kakkonen, T., & Sutinen, E. (2008, October). Evaluation criteria for automatic essay assessment systems – there is much more to it than just the correlation. In *Icce 2008 proceedings* (p. 111-116). Taipei, Taiwan: Asia-Pacific Society for Computers in Education (APSCE). Retrieved 2010.01.02, from `http://www.apsce.net/icce2008/contents/proceeding_0111.pdf` (ICCE Conference on AIED/ITS & Adaptive Learning)

Kies, D. (2010, Dec.). *Evaluating grammar checkers: A comparative ten-year study.* Retrieved 12.5.2011 from `http://papyr.com/hypertextbooks/grammar/gramchek.htm`.

Knight, P. (2001). *A briefing on key concepts* (No. 7). York Science Park, York YO10 5DQ: Learning and Teaching Support Network (LTSN) generic Centre.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to latent semantic analysis. *Discourse Processes*, *25*, 259-284. Available from `http://lsa.colorado.edu/papers/dp1.LSAintro.pdf`

Lang, S. W., & Wilkerson, J. R. (2008, Feb.). Accuracy vs. validity, consistency vs. reliability, and fairness vs. absence of bias: A call for quality. In *Annual meeting of the american association of colleges of teacher education (aacte).* New Orleans, LA: American Association of Colleges of Teacher Education.

Lewis, C., & Rieman, J. (1994). *Task-centered user interface design: A practical introduction.* E-Book published at `ftp://ftp.cs.colorado.edu/pub/cs/distribs/clewis/HCI-Design-Book`, copy retrieved 12.4.2011 from `http://hcibib.org/tcuid/tcuid.pdf`.

Liz, H.-L. (2003). Exploring the dynamics of second language writing. In B. Kroll (Ed.), (3rd ed., p. 162-190). New York: Cambridge University Press.

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, *19*(2), 313-330. Available from `http://acl.ldc.upenn.edu/J/J93/J93-2004.pdf`

Margolis, E., & Laurence, S. (1999). Concepts and cognitive science. In E. Margolis & S. Laurence (Eds.), (p. 3-81). Massachusetts: MIT Press.

Margolis, E., & Laurence, S. (2006). Concepts. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Fall 2008 ed.). Stanford University, Stanford, CA 94305: The Metaphysics Research Lab. `http://plato.stanford.edu/archives/fall2008/entries/concepts/`.

McGee, J. (2010, Sep.). *Toolkit for making written material clear and effective, part 7, using readability formulas: A cautionary note.* Department of Health & Human Services, CMS Centers for Medicare & Medicaid Services. Available from `https://www.cms.gov/WrittenMaterialsToolkit/Downloads/ToolkitPart07-11Files.zip`

McGraw, K. O., & Wong, S. P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological Methods*, *1*(1), 30-46. Available from `http://www3.uta.edu/faculty/ricard/COED/McGraw(1996)`

ForminginferencesaboutICCs.pdf

McMillan, J. H. (2000). Fundamental assessment principles for teachers and school administrators. *Practical Assessment, Research & Evaluation*, *7*(8). Available from http://pareonline.net/getvn.asp?v=7&n=8

McTighe, J., & Arter, J. A. (2004). *Glossary of assessment terms.* http://www.jaymctighe.com/glossary.html accessed 1.10.2010.

Mertler, C. A. (2001). Designing scoring rubrics for your classroom. *Practical Assessment, Research & Evaluation*, *7*(25). Available from http://pareonline.net/getvn.asp?v=7&n=25

Miller, G. A. (1995, November). Wordnet: a lexical database for english. *Commun. ACM*, *38*, 39–41. Available from http://doi.acm.org/10.1145/219717.219748

Mitra, S., Dangwal, R., Chatterjee, S., Jha, S., Bisht, R. S., & Kapur, P. (2005). Acquisition of computing literacy on shared public computers: Children and the 'hole in the wall'. *Australasian Journal of Educational Technology 2005*, *21*(3), 407-426. Available from http://www.eric.ed.gov/ERICWebPortal/custom/portlets/recordDetails/detailmini.jsp?_nfpb=true&_&ERICExtSearch_SearchValue_0=EJ847978&ERICExtSearch_SearchType_0=no&accno=EJ847978

Miłkowski, M. (2010). Developing an open-source, rule-based proofreading tool. *Software – Practice and Experience*, *40*, 543-566.

Molich, R., & Nielsen, J. (1990, March). Improving a human-computer dialogue. *Communications of the ACM*, *33*, 338-348.

Moskal, B. M. (2000). Scoring rubrics: What, when and how? *Practical Assessment, Research & Evaluation*, *7*(4). Available from http://pareonline.net/getvn.asp?v=7&n=3

Moskal, B. M., & Leydens, J. A. (2000). Scoring rubric development: Validity and reliability. *Practical Assessment, Research & Evaluation*, *7*. Available from http://pareonline.net/getvn.asp?v=7&n=10

Naber, D. (2003). *A rule-based style and grammar checker.* Unpublished master's thesis, Technische Fakultät, Universität Bielefeld.

Nielson, J. (1993). *Usability engineering.* San Diego, CA: Academic Press.

Nussbaumer, A., Gütl, C., & Neuper, W. (2010, Sep.). A methodology for adaptive competence assessment and learning path creation isac. In *Proceedings of the international conference on interactive computer-aided learning (icl 2010).* Hasselt, Belgium. Available from http://css.uni-graz.at/staff/nussbaumer/pubfiles/CAF2010-ISAC-Compod.pdf

Osellame, J. (2006, April 4). University opts not to turnitin. *The daily princetonian.* Retrieved 31.12.2010 from http://www.dailyprincetonian.com/2006/04/04/15061/print/.

Page, E. B. (2003). Project essay grade: Peg. In M. D. Shermis & J. C. Burstein (Eds.), (p. 39-50). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Palmer, J., Williams, R., & Dreher, H. (2002). Automated essay grading systems applied to a first year university subject: how can we do it better? In *Is2002 proceedings of the informing science and it education conference, cork, ireland*

(p. 1221-1229). Santa Rosa, California: Informing Science Institute. Available from http://proceedings.informingscience.org/IS2002Proceedings/papers/Palme026Autom.pdf

Pearson Education. (n.d.-a). *Intelligent essay assessor^{TM}- FAQ.* Retrieved 10.2.2010 from, http://www.knowledge-technologies.com/papers/IEA_FAQ.html.

Pearson Education. (n.d.-b). *Pearsons's knowledge technologies products - IEA^{TM}sample feedback screen.* Retrieved 8.2.2010 from, http://www.pearsonkt.com/IEAFeedback.shtml.

Pearson Education. (2007). *IAE results.* Retrieved 8.2.2010 from, http://www.pearsonkt.com/cgi-bin/prenhall/phMain.cgi.

Pellegrino, J. W., Chudowsky, N., & Glaser, R. (Eds.). (2001). *Knowing what students know: The science and design of educational assessment.* Washington, DC: National Academy Press. (Committee on the Foundations of Assessment)

Porter, M. F. (2001, October). *Snowball: A language for stemming algorithms.* Retrieved 18.4.2011 from http://snowball.tartarus.org/texts/introduction.html.

Porter, M. F. (2002, September). *The english (porter2) stemming algorithm.* Retrieved 18.4.2011 from http://snowball.tartarus.org/algorithms/english/stemmer.html.

Powers, D. E., Burstein, J. C., Chodorow, M., Fowles, M. E., & Kukich, K. (2002, March). Stumping e-rater: challenging the validity of automated essay scoring. *Computers in Human Behavior*, *18*(2), 103-134.

Pulman, S. G., & Sukkarieh, J. Z. (2005). Automatic short answer marking. In *Edappsnlp 05: Proceedings of the second workshop on building educational applications using nlp* (pp. 9–16). Morristown, NJ, USA: Association for Computational Linguistics. Available from http://portal.acm.org/citation.cfm?id=1609831

Quinlan, T., Higgins, D., & Wolff, S. (2009, January). *Evaluating the construct-coverage of the e-rater[r] scoring engine* (Research Report No. RR-09-01). Princeton, NJ: Educational Testing Service (ETS). Available from http://www.ets.org/Media/Research/pdf/RR-09-01.pdf

Ramsden, P. (1992). *Learning to teach in higher education.* New Fetter Lane, London: Routledge.

Reazon Systems, I. (2010). *Rcampus.* Retrieved 9.2.2010 from http://www.rcampus.com/indexrubric.cfm.

Redish, J. (2000). Readability formulas have even more limitations than klare discusses. *ACM Journal of Computer Documentation*, *24*(3), 132-137.

Redish, J., & Selzer, J. (1985). The place of readability formulas in technical communication. *Technical Communication*, *32*(2), 46-52.

Reiterer, E., Dreher, H., & Gütl, C. (2010, Feb.). Automatic concept retrieval with rubrico. In M. Schumann, L. M. Kolbe, M. H. Breitner, & A. Frerichs (Eds.), *Multikonferenz wirtschaftsinformatik 2010* (p. 3-14). Göttingen, Germany: Universitätsverlag Göttingen. Available from http://webdoc.sub.gwdg.de/univerlag/2010/mkwi/01_management_und_methoden/anw._konz_ont_mod/01_automatic_concept_retrieval_with_rubrico.pdf

Rich, C., & Wang, Y. (2010, june). Online formative assessment using automated essay

scoring technology in china and u.s. - two case studies. In *Education technology and computer (icetc), 2010 2nd international conference on* (Vol. 3, p. V3-524 -V3-528).

Richardson, E. (2002). *Essay questions.* Retrieved 11.7.2010 from `http://web.utk.edu/~mccay/apdm/essay/essay.pdf`.

Rudner, L., & Gagne, P. (2001). An overview of three approaches to scoring written essays by computer. *Practical Assessment, Research & Evaluation*, *7*(26). Available from `http://pareonline.net/getvn.asp?v=7&n=26`

Rudner, L. M., Garcia, V., & Welch, C. (2006, March). An evaluation of intellimetric™essay scoring system. *Journal of Technology, Learning and Assessment*, *4*(4). Available from `http://escholarship.bc.edu/jtla/vol4/4/`

Saunders, P. I. (1991, November). Primary trait scoring: A direct assessment option for educators. In *National council of teachers of english annual convention.* Denver, CO: National Council of Teachers of English. Available from `http://www.eric.ed.gov/ERICWebPortal/contentdelivery/servlet/ERICServlet?accno=ED444624`

Schilit, B. N., Golovchinsky, G., & Price, M. N. (1998). Beyond paper: supporting active reading with free form digital ink annotations. In C. Karat, A. Lund, J. Coutaz, & J. Karat (Eds.), (p. 249-256). New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Scouller, K. (1997, July). Students' perceptions of three assessment methods: Assignment essay, multiple choice question examination, short answer examination. In *Herdsa'97 conference: Advancing international perspectives* (p. 646-653). Adelaide, NSW: HERDSA. Available from `http://www.herdsa.org.au/wp-content/uploads/conference/1997/scoull01.pdf`

Shaw, S. (2008). Essay marking on-screen: implications for assessment validity. *E-Learning and Digital Media*, *5*(3).

Shepard, L. A. (2000, October 01). The role of assessment in a learning culture. *Educational Researcher*, *29*(7), 4-14.

Shermis, M. D., & Burstein, J. C. (Eds.). (2003). *Automated essay scoring: A cross-disciplinary perspective.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

Shermis, M. D., Mzumara, H. R., Olson, J., & Harrington, S. (2001, June). Online grading of student essays: Peg goes on the world wide web. *Assessment & Evaluation in Higher Education*, *26*(3), 247-259. Available from `http://www.cblt.soton.ac.uk/principles03/pdf/On-lineGrad_N_PEG.pdf`

Shneiderman, B., Grinstein, G., Kobsa, A., Plaisant, C., & Stasko, J. T. (2003). Which comes first, usability or utility? In G. Turk, J. J. van Wiljk, & R. Moorhead (Eds.), *14th ieee visualization 2003* (p. 112). Los Alamitos, CA, USA: IEEE Computer Society. (14th IEEE Visualization 2003 conference)

Shortis, M., & Burrows, S. (2009, Nov.). A review of the status of online, semi-automated marking and feedback systems. In J. Milton, C. Hall, J. Lang, G. Allan, & M. Nomikoudis (Eds.), *Atn assessment conference 2009: Assessment in different dimensions.* Melbourne, Australia: Learning and Teaching Unit, RMIT University. Available from `http://emedia.rmit.edu.au/conferences/index.php/`

ATNAC/ATNAC09/paper/view/205/46

Sim, G., Holifield, P., & Brown, M. (2004). Implementation of computer assisted assessment: lessons from the literature. *Association for Learning Technology Journal*, *12*(3), 215-229. Available from http://repository.alt.ac.uk/608/

Snow, R., Prakash, S., Jurafsky, D., & Ng, A. Y. (2007, June). Learning to merge word senses. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll)* (pp. 1005–1014). Prague, Czech Republic: Association for Computational Linguistics. Available from http://ai.stanford.edu/~rion/papers/wordsenses_emnlp07.pdf

Stecher, B. M., Rahn, M. L., Ruby, A., Alt, M. N., & Robyn, A. (1997). *Using alternative assessments in vocational education.* Santa Monica, CA: Rand.

Stemler, S. E. (2004). A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Practical Assessment, Research & Evaluation*, *9*(4). Available from http://PAREonline.net/getvn.asp?v=9&n=4

Stiggins, R. (2007, May). Assessment through the student's eyes. *Educational Leadership*, *64*(8), 22-26. Available from http://www.ascd.org/publications/educational_leadership/may07/vol64/num08/Assessment_Through_the_Student's_Eyes.aspx

Stiggins, R. J., Arter, J. A., Chappuis, J., & Chappuis, S. (2004). *Classroom assessment for student learning* (R. L. Marcum, Ed.). Portland, Oregon: Assessment Training Institute Inc.

Teferra, D. (2001). Academic dishonesty in african universities–trends, challenges, and repercussions: An ethiopian case study. *International Journal of Educational Development*, *21*(2), 163 - 178. Available from http://www.sciencedirect.com/science/article/B6VD7-4292H1J-7/2/1d4bce2b84aabe91af2f0f720030883b

Toutanova, K., & Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the joint sigdat conference on empirical methods in natural language processing and very large corpora (emnlp/vlc-2000)* (p. 63-70).

Valenti, S., Neri, F., & Cucchiarelli, A. (2003). An overview of current research on automated essay grading. *Journal of Information Technology Education*, *2*, 319-330. Available from http://jite.org/documents/Vol2/v2p319-330-30.pdf

Vantage Learning. (n.d.). *Demo center :: My access!® school edition - product overview.* Retrieved 9.2.2010 from http://www.vantagelearning.com/school/demos/demos_myaccess_overview_skin.

Vantage Learning. (2005). *How intellimetric^{TM}works.* Retrieved 8.1.2010 from http://www.vantagelearning.com/docs/intellimetric/IM_How_IntelliMetric_Works.pdf.

Wang, J., & Brown, M. S. (2007, October). Automated essay scoring versus human scoring: A comparative study. *Journal of Technology, Learning and Assessment*, *6*(2). Available from http://escholarship.bc.edu/jtla/vol6/2/

Warschauer, M., & Ware, P. (2006). Automated writing evaluation: Defining the classroom research agenda. *Language Teaching Research*, *10*(2), 157-180. Available

from http://www.gse.uci.edu/person/warschauer_m/docs/AWE.pdf

Weigle, S. C. (1999). Investigating rater/prompt interactions in writing assessment: Quantitative and qualitative approaches. *Assessing Writing*, *6*(2), 145 - 178. Available from http://www.sciencedirect.com/science/article/B6VT8-42DXWV4-2/2/24c7e196cf2b4b5b16b3bbcd4a950681

Weigle, S. C. (2002). *Assessing writing*. Cambridge: Cambridge University Press.

Weimer, M. (2002). *Learner-centered teaching: Five key changes to practice*. San Francisco, CA 94103-1741: Jossey-Bass.

Weinhofer, J. (2010). *Extraktion semantisch relevanter daten*. Unpublished master's thesis, Technische Unversität Graz.

White, T. (2004, Sep.). *Can't beat jazzy - introducing the java platform's jazzy new spell checker api*. Retrieved 20.11.2010 from http://www.ibm.com/developerworks/java/library/j-jazzy/.

Wikipedia. (2010a, Dec.). *Flesch–kincaid readability test*. Retrieved 17.12.2010 from http://en.wikipedia.org/w/index.php?title=Flesch%E2%80%93Kincaid_readability_test&oldid=402715098. (In Wikipedia, The Free Encyclopedia)

Wikipedia. (2010b, January). *Inter-rater reliability*. Retrieved 15.1.2010 from http://en.wikipedia.org/w/index.php?title=Inter-rater_reliability&oldid=336681105. (In Wikipedia, The Free Encyclopedia)

Willet, P. (2006). The porter stemming algorithm: then and now. *Program: electronic library and information systems*, *40*(3), 219-223. Available from http://eprints.whiterose.ac.uk/1434/01/willettp9_PorterStemmingReview.pdf

Williams, R. (2006). The power of normalised word vectors for automatically grading essays. *Issues in Informing Science and Information Technology*, *3*, 721. Available from http://informingscience.org/proceedings/InSITE2006/IISITWill155.pdf

Williams, R. (2007, February). A computational effective document semantic representation. In *Ieee international conference on digital ecosystems and technologies, ieee dest 2007, cairns* (p. 410-415). Cairns, Australia.

Williams, R., & Dreher, H. (2004). Automatically grading essays with markit©. *Issues in Informing Science and Information Technology*, *1*, 693-700.

Winters, T., & Payne, T. (2005). What do students know?: an outcomes-based assessment system. In (p. 165-172). New York, NY, USA: ACM.

A. Acronyms

AEG Automated Essay Grading. 11, 21, 22, 43, 47, 48

AES Automated Essay Scoring. 21

AWE Automated Writing Evaluation. 21

BSD Berkeley Software Distribution. 65, 141

CSS Cascading Style Sheets. 68

GMAT Graduate Management Admission Test. 21, 41, 46

GPL GNU General Public License. 65, 66, 141

GUI Graphical User Interface. 52, 53, 58–60, 63–65, 67–69, 77, 80, 92, 94–97

HTML HyperText Markup Language. 61, 66, 68, 77

IDE Integrated Development Environment. 63, 94, 96

IEA™ Intelligent Essay Assessor™. 40, 46

JPF Java Plug-in Framework. 64

LGPL GNU Lesser General Public License. 61, 64, 65, 69, 70, 141

LMS Learning Management System. 46, 60, 62, 75, 85, 94, 98, 99

LSA Latent Semantic Analysis. 39, 40, 44

NLP Natural Language Processing. 41, 59

PEG Project Essay Grade™. 38, 47

POS Part of Speech. 59, 63, 65

SFTA Short Free Text Answers. 45

SOA Service-Oriented Architecture. 64, 99

SQL Structured Query Language. 61, 97

XML Extensible Markup Language. 61–63, 65–68, 70, 72, 77, 89

B. List of Figures

C. List of Tables

D. CD-ROM

CD-ROM

E. Essay XML Format

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document>
 <statistics>
  <wordcount>#</wordcount>
  <charcount>#</charcount>
  <whitespacecount>#</whitespacecount>
  <headingcount>#</headingcount>
  <paragraphcount>#</paragraphcount>
  <sentencecount>#</sentencecount>
  <commacount>#</commacount>
  <parenthesiscount>#</parenthesiscount>
  <otherinterpunctuationcount>#</otherinterpunctuationcount>
  <punctuationmarkcount>#</punctuationmarkcount>
 </statistics>
 <cssstylelist>
   <cssstyle id="criterionid">cssstyle</cssstyle>
 </cssstylelist>
 <text>
  <heading endchar="#" id="heading#" number="#" startchar="#">
    <word endchar="#" id="word#" lowercase="text" startchar="#"
       stemmed="text" baseform="text" pos="identifier">text</word>
  </heading>
  <paragraph endchar="#" id="paragraph#" number="#" startchar="#">
    <sentence endchar="#" id="sentence#" number="#" original="text"
        startchar="#">
     <word endchar="#" id="word#" lowercase="text" startchar="#"
        stemmed="text" baseform="text" pos="identifier">text</word>
     <interpunctuation endchar="#" id="interpunctuation#"
        startchar="#">symbol</interpunctuation>
     <markup class="identifier">
       <word endchar="#" id="word#" lowercase="text" startchar="#"
          stemmed="text" baseform="text" pos="identifier">text</word>
       <comma endchar="#" id="comma#" startchar="#">,</comma>
       <word endchar="#" id="word#" lowercase="text" startchar="#"
          stemmed="text" baseform="text" pos="identifier">text</word>
     </markup>
     <punctuationmark endchar="#" id="punctuationmark#"
        startchar="#">symbol</punctuationmark>
    </sentence>
  </paragraph>
 </text>
</document>
```

Parsed example essay


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document>
<statistics>
<wordcount>94</wordcount>
<charcount>585</charcount>
<whitespacecount>109</whitespacecount>
<headingcount>1</headingcount>
<paragraphcount>2</paragraphcount>
<sentencecount>3</sentencecount>
<commacount>4</commacount>
<parenthesiscount>0</parenthesiscount>
<otherinterpunctuationcount>33</otherinterpunctuationcount>
<punctuationmarkcount>3</punctuationmarkcount>
</statistics>
<cssstylelist/>
<text>
<heading endchar="28" id="heading0" main="true" number="0" startchar="0">
<word baseform="Business" endchar="8" id="word0" lowercase="business" pos="NNP" startcha
<word baseform="situation" endchar="18" id="word1" lowercase="situation" pos="NN" start
<word baseform="analysis" endchar="27" id="word2" lowercase="analysis" pos="NN" startcha
</heading>
<paragraph endchar="292" id="paragraph0" number="0" startchar="28">
<sentence endchar="291" id="sentence0" number="0" original="In this argument Aura's sal
<word baseform="in" endchar="30" id="word3" lowercase="in" pos="IN" startchar="28" stemm
<word baseform="this" endchar="35" id="word4" lowercase="this" pos="DT" startchar="31"
<word baseform="argument" endchar="44" id="word5" lowercase="argument" pos="NN" startcha
<word baseform="aura'" endchar="51" id="word6" lowercase="aura's" pos="VBZ" startchar="
<word baseform="sale" endchar="57" id="word7" lowercase="sales" pos="NNS" startchar="52
<word baseform="director" endchar="66" id="word8" lowercase="director" pos="NN" startch
<word baseform="rely" endchar="73" id="word9" lowercase="relies" pos="VBZ" startchar="6
<word baseform="on" endchar="76" id="word10" lowercase="on" pos="IN" startchar="74" stem
<word baseform="certain" endchar="84" id="word11" lowercase="certain" pos="JJ" startcha
<word baseform="anecdotal" endchar="94" id="word12" lowercase="anecdotal" pos="JJ" star
<word baseform="evidence" endchar="103" id="word13" lowercase="evidence" pos="NN" start
<word baseform="about" endchar="109" id="word14" lowercase="about" pos="IN" startchar="
<word baseform="one" endchar="113" id="word15" lowercase="one" pos="CD" startchar="110"
<word baseform="other" endchar="119" id="word16" lowercase="other" pos="JJ" startchar="
<word baseform="" endchar="127" id="word17" lowercase="company" pos="" startchar="120"
<comma endchar="128" id="comma0" startchar="127">,</comma>
<word baseform="as" endchar="130" id="word18" lowercase="as" pos="RB" startchar="128" st
<word baseform="well" endchar="135" id="word19" lowercase="well" pos="RB" startchar="13
<word baseform="as" endchar="138" id="word20" lowercase="as" pos="IN" startchar="136" st
<word baseform="certain" endchar="146" id="word21" lowercase="certain" pos="JJ" startch
<word baseform="statistics" endchar="157" id="word22" lowercase="statistics" pos="NNS"
<word baseform="about" endchar="163" id="word23" lowercase="about" pos="IN" startchar="
<word baseform="general" endchar="171" id="word24" lowercase="general" pos="JJ" startch
<word baseform="trend" endchar="178" id="word25" lowercase="trends" pos="NNS" startchar
<word baseform="among" endchar="184" id="word26" lowercase="among" pos="IN" startchar="
<word baseform="large" endchar="190" id="word27" lowercase="large" pos="JJ" startchar="
<word baseform="" endchar="200" id="word28" lowercase="companies" pos="" startchar="191
```

<comma endchar="201" id="comma1" startchar="200">,</comma>
<word baseform="to" endchar="203" id="word29" lowercase="to" pos="TO" startchar="201" st
<word baseform="convince" endchar="212" id="word30" lowercase="convince" pos="VB" startc
<word baseform="we" endchar="215" id="word31" lowercase="us" pos="PRP" startchar="213" s
<word baseform="of" endchar="218" id="word32" lowercase="of" pos="IN" startchar="216" st
<word baseform="the" endchar="222" id="word33" lowercase="the" pos="DT" startchar="219"
<word baseform="merit" endchar="229" id="word34" lowercase="merits" pos="NNS" startchar
<word baseform="of" endchar="232" id="word35" lowercase="of" pos="IN" startchar="230" st
<word baseform="enrol" endchar="242" id="word36" lowercase="enrolling" pos="VBG" startcl
<word baseform="certain" endchar="250" id="word37" lowercase="certain" pos="JJ" startch
<word baseform="ABC" endchar="254" id="word38" lowercase="abc" pos="NNP" startchar="251
<word baseform="employee" endchar="264" id="word39" lowercase="employees" pos="NNS" sta
<word baseform="in" endchar="267" id="word40" lowercase="in" pos="IN" startchar="265" st
<word baseform="the" endchar="271" id="word41" lowercase="the" pos="DT" startchar="268"
<word baseform="SureSale" endchar="280" id="word42" lowercase="suresale" pos="NNP" star
<word baseform="seminar" endchar="288" id="word43" lowercase="seminar" pos="NN" startch
<punctuationmark endchar="289" id="punctuationmark0" startchar="288">.</punctuationmark
</sentence>
</paragraph>
<paragraph endchar="627" id="paragraph1" number="1" startchar="290">
<sentence endchar="490" id="sentence1" number="1" original="Turning first to the anecdo
<word baseform="turn" endchar="297" id="word44" lowercase="turning" pos="VBG" startchar=
<word baseform="first" endchar="303" id="word45" lowercase="first" pos="RB" startchar="
<word baseform="to" endchar="306" id="word46" lowercase="to" pos="TO" startchar="304" st
<word baseform="the" endchar="310" id="word47" lowercase="the" pos="DT" startchar="307"
<word baseform="anecdotal" endchar="320" id="word48" lowercase="anecdotal" pos="JJ" sta
<word baseform="" endchar="329" id="word49" lowercase="evidence" pos="" startchar="321"
<comma endchar="330" id="comma2" startchar="329">,</comma>
<word baseform="the" endchar="333" id="word50" lowercase="the" pos="DT" startchar="330"
<word baseform="director" endchar="342" id="word51" lowercase="director" pos="NN" startc
<word baseform="assume" endchar="350" id="word52" lowercase="assumes" pos="VBZ" startch
<word baseform="too" endchar="354" id="word53" lowercase="too" pos="RB" startchar="351"
<word baseform="hastily" endchar="362" id="word54" lowercase="hastily" pos="RB" startch
<word baseform="that" endchar="367" id="word55" lowercase="that" pos="IN" startchar="36
<word baseform="the" endchar="371" id="word56" lowercase="the" pos="DT" startchar="368"
<word baseform="SureSale" endchar="380" id="word57" lowercase="suresale" pos="NNP" star
<word baseform="seminar" endchar="388" id="word58" lowercase="seminar" pos="NN" startch
<interpunctuation endchar="390" id="interpunctuation28" startchar="389">-</interpunctua
<interpunctuation endchar="391" id="interpunctuation29" startchar="390">-</interpunctua
<word baseform="rather" endchar="397" id="word59" lowercase="rather" pos="RB" startchar=
<word baseform="than" endchar="402" id="word60" lowercase="than" pos="IN" startchar="39
<word baseform="some" endchar="407" id="word61" lowercase="some" pos="DT" startchar="40
<word baseform="other" endchar="413" id="word62" lowercase="other" pos="JJ" startchar="
<word baseform="phenomenon" endchar="424" id="word63" lowercase="phenomenon" pos="NN" s
<interpunctuation endchar="426" id="interpunctuation30" startchar="425">-</interpunctua
<interpunctuation endchar="427" id="interpunctuation31" startchar="426">-</interpunctua
<word baseform="be" endchar="430" id="word64" lowercase="was" pos="VBD" startchar="427"
<word baseform="responsible" endchar="442" id="word65" lowercase="responsible" pos="JJ"
<word baseform="for" endchar="446" id="word66" lowercase="for" pos="IN" startchar="443"
<word baseform="the" endchar="450" id="word67" lowercase="the" pos="DT" startchar="447"
<word baseform="increase" endchar="459" id="word68" lowercase="increase" pos="NN" startc
<word baseform="in" endchar="462" id="word69" lowercase="in" pos="IN" startchar="460" st

<word baseform="TechAide's" endchar="473" id="word70" lowercase="techaide's" pos="NNP"
<word baseform="total" endchar="479" id="word71" lowercase="total" pos="JJ" startchar="
<word baseform="sale" endchar="485" id="word72" lowercase="sales" pos="NNS" startchar="
<punctuationmark endchar="486" id="punctuationmark1" startchar="485">.</punctuationmark
</sentence>
<sentence endchar="623" id="sentence2" number="2" original="Perhaps the increase simply
<word baseform="perhaps" endchar="494" id="word73" lowercase="perhaps" pos="RB" startch
<word baseform="the" endchar="498" id="word74" lowercase="the" pos="DT" startchar="495"
<word baseform="increase" endchar="507" id="word75" lowercase="increase" pos="NN" start
<word baseform="simply" endchar="514" id="word76" lowercase="simply" pos="RB" startchar
<word baseform="reflect" endchar="524" id="word77" lowercase="reflected" pos="VBD" star
<word baseform="general" endchar="532" id="word78" lowercase="general" pos="JJ" startch
<word baseform="economic" endchar="541" id="word79" lowercase="economic" pos="JJ" start
<word baseform="or" endchar="544" id="word80" lowercase="or" pos="CC" startchar="542" st
<word baseform="" endchar="551" id="word81" lowercase="supply" pos="" startchar="545" st
<interpunctuation endchar="552" id="interpunctuation32" startchar="551">-</interpunctua
<word baseform="" endchar="558" id="word82" lowercase="demand" pos="" startchar="552" st
<word baseform="" endchar="564" id="word83" lowercase="trends" pos="" startchar="558" st
<comma endchar="565" id="comma3" startchar="564">,</comma>
<word baseform="" endchar="567" id="word84" lowercase="or" pos="" startchar="565" stemm
<word baseform="" endchar="569" id="word85" lowercase="a" pos="" startchar="568" stemme
<word baseform="" endchar="577" id="word86" lowercase="misstep" pos="" startchar="570"
<word baseform="" endchar="580" id="word87" lowercase="on" pos="" startchar="578" stemm
<word baseform="" endchar="584" id="word88" lowercase="the" pos="" startchar="581" stem
<word baseform="" endchar="589" id="word89" lowercase="part" pos="" startchar="585" sten
<word baseform="" endchar="592" id="word90" lowercase="of" pos="" startchar="590" stemm
<word baseform="" endchar="603" id="word91" lowercase="techaide's" pos="" startchar="59
<word baseform="" endchar="609" id="word92" lowercase="chief" pos="" startchar="604" ste
<word baseform="" endchar="620" id="word93" lowercase="competitor" pos="" startchar="61
<punctuationmark endchar="621" id="punctuationmark2" startchar="620">.</punctuationmark
</sentence>
</paragraph>
</text>
</document>

F. XSL Stylesheet

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
<html>
<head>
<style>
body {margin:5 5 5 50;}
p {margin: 10 0 0 0; line-height: 1.1em;}
* {font-family:'Times New Roman',Times,serif;}
div.markup{display:inline; border-bottom:2px solid blue;}
<xsl:for-each select="document/cssstylelist/cssstyle">
 <xsl:text><xsl:value-of select="."/>
 </xsl:text>
</xsl:for-each>
</style>
</head>
<body>
 <xsl:for-each select="document/text">
 <xsl:for-each select="child::*">
 <xsl:if test="name()='heading'">
  <h1>
  <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
  <xsl:attribute name="class">heading <xsl:value-of select="@class"/>
    </xsl:attribute>
   <xsl:for-each select="child::*">
    <xsl:if test="name()='word'">
     <xsl:text> </xsl:text>
     <span>
     <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
     <xsl:attribute name="class">word <xsl:value-of select="@class"/>
       </xsl:attribute>
     <xsl:value-of select="." />
     </span>
    </xsl:if>
    <xsl:if test="name()='punctuationmark'">
     <span>
     <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
     <xsl:attribute name="class">punctuationmark
       <xsl:value-of select="@class"/></xsl:attribute>
     <xsl:value-of select="." />
     </span>
    </xsl:if>
    <xsl:if test="name()='comma'">
     <span>
     <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
```

```
 <xsl:attribute name="class">comma <xsl:value-of select="@class" />
   </xsl:attribute>
 <xsl:value-of select="." />
 </span>
</xsl:if>
<xsl:if test="name()='interpunctuation'">
  <span>
  <xsl:attribute name="id"><xsl:value-of select="@id" /></xsl:attribute>
  <xsl:attribute name="class">comma <xsl:value-of select="@class" />
    </xsl:attribute>
  <xsl:value-of select="." />
  </span>
 </xsl:if>
 <xsl:if test="name()='markup'">
  <span>
  <xsl:attribute name="class">markup <xsl:value-of select="@class" />
    </xsl:attribute>
  <xsl:for-each select="child::*">
   <xsl:if test="name()='word'">
    <xsl:text> </xsl:text>
      <span>
      <xsl:attribute name="id"><xsl:value-of select="@id" /></xsl:attribute>
      <xsl:attribute name="class">word <xsl:value-of select="@class" />
        </xsl:attribute>
      <xsl:value-of select="." />
      </span>
     </xsl:if>
     <xsl:if test="name()='comma'">
      <span>
      <xsl:attribute name="id"><xsl:value-of select="@id" /></xsl:attribute>
      <xsl:attribute name="class">comma <xsl:value-of select="@class" />
        </xsl:attribute>
      <xsl:value-of select="." />
      </span>
     </xsl:if>
     <xsl:if test="name()='interpunctuation'">
      <span>
      <xsl:attribute name="id"><xsl:value-of select="@id" /></xsl:attribute>
      <xsl:attribute name="class">comma <xsl:value-of select="@class" />
        </xsl:attribute>
      <xsl:value-of select="." />
      </span>
     </xsl:if>
             <xsl:if test="name()='punctuationmark'">
      <span>
      <xsl:attribute name="id"><xsl:value-of select="@id" /></xsl:attribute>
      <xsl:attribute name="class">punctuationmark
        <xsl:value-of select="@class" /></xsl:attribute>
      <xsl:value-of select="." />
      </span>
    </xsl:if>
    </xsl:for-each>
```

```
      </span>
     </xsl:if>
    </xsl:for-each>
</h1>
</xsl:if>
<xsl:if test="name()='paragraph'">
 <p>
 <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
 <xsl:attribute name="class">paragraph <xsl:value-of select="@class"/>
   </xsl:attribute>
 <xsl:for-each select="sentence">
   <span>
 <xsl:for-each select="child::*">
  <xsl:if test="name()='word'">
   <xsl:text> </xsl:text>
   <span>
   <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
   <xsl:attribute name="class">word <xsl:value-of select="@class"/>
     </xsl:attribute>
   <xsl:value-of select="." />
   </span>
  </xsl:if>
  <xsl:if test="name()='punctuationmark'">
   <span>
   <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
   <xsl:attribute name="class">punctuationmark
     <xsl:value-of select="@class"/></xsl:attribute>
   <xsl:value-of select="." />
   </span>
  </xsl:if>
  <xsl:if test="name()='comma'">
   <span>
   <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
   <xsl:attribute name="class">comma <xsl:value-of select="@class"/>
     </xsl:attribute>
   <xsl:value-of select="." />
   </span>
  </xsl:if>
  <xsl:if test="name()='interpunctuation'">
    <span>
    <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
    <xsl:attribute name="class">comma <xsl:value-of select="@class"/>
      </xsl:attribute>
    <xsl:value-of select="." />
    </span>
   </xsl:if>
   <xsl:if test="name()='markup'">
    <span>
    <xsl:attribute name="class">markup <xsl:value-of select="@class"/>
      </xsl:attribute>
    <xsl:for-each select="child::*">
     <xsl:if test="name()='word'">
```

```
        <xsl:text> </xsl:text>
          <span>
          <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
          <xsl:attribute name="class">word <xsl:value-of select="@class"/>
            </xsl:attribute>
          <xsl:value-of select="." />
          </span>
         </xsl:if>
         <xsl:if test="name()='comma'">
          <span>
          <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
          <xsl:attribute name="class">comma <xsl:value-of select="@class"/>
            </xsl:attribute>
          <xsl:value-of select="." />
          </span>
         </xsl:if>
         <xsl:if test="name()='interpunctuation'">
          <span>
          <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
          <xsl:attribute name="class">comma <xsl:value-of select="@class"/>
            </xsl:attribute>
          <xsl:value-of select="." />
          </span>
         </xsl:if>
                 <xsl:if test="name()='punctuationmark'">
          <span>
          <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
          <xsl:attribute name="class">punctuationmark
            <xsl:value-of select="@class"/></xsl:attribute>
          <xsl:value-of select="." />
          </span>
         </xsl:if>
         </xsl:for-each>
        </span>
       </xsl:if>
      </xsl:for-each>
      </span>
     <xsl:text> </xsl:text>
    </xsl:for-each>
    </p>
    </xsl:if>
   </xsl:for-each>
  </xsl:for-each>
</body></html>
</xsl:template>
</xsl:stylesheet>
```

G. Grammar Errors Tests

The data of the twenty most frequent errors comes from the original article published by R. J. Connors and Lunsford (1988), as cited in the grammar checker comparison by Kies (2010).

1. No comma after introductory element
   After we watched the movie we went over to the pizza joint for a bite to eat.

2. Vague pronoun reference
   John Smith reported the problem to Bob Adams, and he corrected it immediately.

3. No comma in compound sentence
   John Smith reported the problem to Bob Adams and Adams corrected it immediately.

4. Wrong word
   Building a new fiber optic network will take less dollars then originally suspected.

5. No comma in nonrestrictive element
   Contemporary animal rights groups who are using the same strategies that the civil rights movement developed in the 1960s find that non-violent civil disobedience is effectively drawing attention to their cause.

6. Wrong or missing inflected endings
   Teacher representatives in the United States' could have wrote the book themselves.

7. Wrong or missing preposition
   The man to whom you were speaking to was the curator on the exposition.

8. Comma splice
   In 1980, Reagan promised to balance the federal budget, however, by 1988, the federal deficit was the largest in U. S. history — until now.

9. Possessive apostrophe error
   The Presidents speech was punctuated by enthusiastic applause from both sides of the aisle.

10. Tense shift
    John was edgy all day. He challenged anything that anyone said to him. Once at a meeting, John looked right at me and he says, "What do you mean by that?"

11. Unnecessary shift in person
    Students should register for classes early if you want to ensure that you get a good schedule.

12. Sentence fragment
    Publishers today are at a loss trying to cope with the new trends in information technology. Which is not surprising actually.

13. Wrong tense or verb form
    The market has responding favorably to the surprisingly strong earnings that the high-tech sector reports yesterday.

14. Subject-verb agreement
    The President as well as his advisors were in meetings all weekend at Camp David. Neither the President nor his advisors is commenting on the status of the negotiations.

15. Lack of comma in a series
    Forget what Wheaties tells you; the real breakfast of champions is pizza, beer and eggs.

16. Pronoun agreement error
    Everyone should register early to make sure that they get the classes they want.

17. Unnecessary comma with restrictive element
    Several of the guys on the team were here at the time, but the guy, who is talking to the police, is the only witness to the accident.

18. Run-on or fused sentence
    Courses in entrepreneurial businesses are increasingly popular however few faculty members are qualified to teach them.

19. Dangling or misplaced modifier
    While smoking a cigarette, the coffee in his cup had grown cold. He says often he smokes cigarettes.

20. Its/it's error
    Its commonly assumed that any bear will defend it's offspring by attacking any human that wanders too near a cub.

H. GUI Review Protocols

In the following protocols lines starting with "O:" are prompts and hints given by the observer during the review sessions. All other lines have been said by the test users. Lines in parenthesis are silent notes by the observer, written during the live protocol taking. These have been taken to ease the analysis by providing some context.

Think aloud protocol 12.5.2011 GUI Test User 1

I'm not sure what to enter. Can I choose a password?

O: The password is .... for the login screen

(User reads the introduction help screen)

There is a typo in the text, designed is misspelt

(User performs the step in the first assignment wizard)

Ok, I find it a bit unclear. The introduction says assignment, then rubric, then import student essays...but now it asks me to add essays. (assignment creation wizard, first run)

(user is adding some essays) Am I too slow? The double-click needs to be really fast.

(user changes GUI layout)

I don't know what these tabs do? (essay content vs. original essay) What is the difference between essay content and original essay? That is not intuitive to me

That's a bit odd...I don't see the tab any more. The title vanished...

(user switches tabs assignment list and assignment details)

Here it wasn't clear what I can search for...

O: Please tell all your thoughts about using the GUI, not only things you don't like.

(user is exploring configuration screen)

Hm...I can't change anything here. Oh, here there is a bit.

I find the layout quite clear. Assignments on the left side, on the right the essay is displayed. And here I can select or import the rubric.

O: Please now create a rubric for the assignment and use it.

I don't know which criteria are available when I first use it.
But I should give the rubric a title now.

I like that it brings me to the settings now (plugins not activated yet)
Oh, that's very good. I get right back where I was (after plugin activation back to rubric creation dialog)

Hm...I don't know...(is looking at example essays for rubric)
The button was not clear for me. With the dialog its becoming clear (adding example essays)
Should I do something with it now?

I'm a surprised...I can add more than one plugin to the rubric
Maybe I don't know enough about the system function

That's odd. I had added an positive example but now its gone (user discover a bug in the GUI)
It's still there if I open the dialog

(user notices the orange icon, read tooltip that rubric is not ready to use yet)
Nothing happens (user tries to double click criterion in list)

Setting up the criteria is quite clear after you found the way to edit them

(user finished rubric and selects it)

Hmm...that doesn't work (user tries to unintentionally manually rate the essay)

(user clicks on evaluate)
Is the system still busy? Do I have to wait?

O: No. You can continue work

Ah..now I see a result

O: Please add another criterion to the rubric.

Okay...hm...I can't edit it (user open the rubric and clicked the icon to view it)
Hm..I can't do anything. I can't change it. Does it work to add a second criterion?

O: Go back one step.

I'm not allowed. Why I'm I not allowed to do it?
Ok, then not.

O: It is possible.

Hm..oh there is an unlock. I try it.
Ok, now it makes sense. It tells me it deletes the results. That's clear now. When I change the rubric the old results will be invalid. But I didn't grasp that in the screen before. Maybe I didn't see it.

I add the essay length criterion.
I don't know. Is the length in characters or in words?
O: In words.

Hm, I can add a level. Done.

I'll another one. Readability. (Reading explanation)
Done.

And now? What does cancel do? I'm not sure. I'm not sure if it throws away everything.
I think that should be renamed. (Cancel button in rubric selection dialog)

(evaluates again)
The automatic results are clear. Its easy to see the results. And I know where they come from, I set it up earlier.

What comes into the space on the right? Other criteria when I add more? (referring to the empty space in the compact rubric view).

Ah, I can show the whole rubric too. But there is not enough space for it. Maybe...
(user undocks the rubric view).
Nice. I can put in on the second screen

(user clicks on criterion. Level display pops up. Unintentionally sets manual rating).

(open criterion again)

Hm..there are two borders now. I'm trying to figure out what they mean..and also there is a manual and automatic rating there now.
Is the automatic always a dashed border?
Still don't know where the manual rating is coming from.

O: You made it.

Really? I don't know how. Oh. When I open it and click it, it vanishes. I only see the change when I open it the next time.
Can I delete the manual rating?
Ok, I can just click on the automatic one
. That a bit tricky. Ok, I find it acceptable. Its an expert tool.
The manual rating overrides the automatic one. The automatic becomes meaningless.

O: Choose a longer essay. Can you tell me what is wrong in it?

There is a list of spelling errors below it. When I click it the errors is highlighted in the text.
What does highlighting do? Ah, I can mark all the errors at once.

O: If you think the automatic results isn't correct for spelling. Can you alter it?

Yes, I think so. There was something...(user goes back to edit the rubric).
Here I can mark words as correct. But the list is empty.
Maybe if I add an example essays. I need to add the essay again?
Hm, the example essays are ambiguous to me now.

O: Check the essay length criterion.

Ok, it now tells me the average length. I think the example essays are there to train the criteria

(Goes back to spelling. Marks the erroneous words as correct)
Not sure if it takes them only from the positive, negative or both examples.
Is there another way?

O: Yes.

(goes back to essays. evaluates)
(Selects the error in the list).
It highlights...maybe...ok, done (right clicked the word in the list)
(Tries to select multiple words at once)

It doesn't like several at once.

It's not sorted, the results. I want to sort the results. It seems I can't do that.

O: That's correct.

O: When you mark a word as correct does it effect the other essays? Correct one word now. Remember all essays have been evaluated already.

I don't know. I need to try it. I does not apply it - according to the timestamp it's only applied it to the current one.
Oh, that's the submission time.

O: Thanks for your participation in the test. Do you have any remarks now?

How long did it take us? 30 minutes is enough. I think I'm confident now to use the program. If you have seen it once you can work with it. Trying it out once is enough. Not everything is self explanatory. Especially the example essays. I now also like that it closes the criterion when I manually rate it - its faster.

O: Should there be more explanations or help available?

I thinks its okay. You need one training session but then you can work with it. More help would annoy me when using it daily, especially assistants.

O: Thank you for your help.

Protocol 12.5.2011 GUI Test User 2

(user steps through installation dialog)

Both things are marked as ok, I can continue

O: The password is .... for the login screen

The data is not necessary, I skip it (refering to user setup)

(user reads instructions)

Hm, what do I enter for the title? Ok, it just seems to be something like a heading

Course number is not necessary
What can I choose here? Oh, a calendar
I'll add one week for correcting the assignment, as usual in school (submission deadline)

(tries to add essay, successfully adds one, come to main screen)

I assume I have to add something here. Do I do the same as before?
(adds another essay)
(starts reading the essay on the right side)
I have added an essay but I'm not sure what is displayed on the right side here.
(As an extremely short one was selected user needs time to recognize it as the essay content)

O: Please add some more essays.

I'll add an essay for the same student I assume
(dialog pops up that an essay exists)
Ok, that doesn't work that I add an essay for this student. I need to decide to either keep the previous one or overwrite it with the current
I only have the possibility to add an essay for another student
(user deletes an essay and adds another)
So I can also change the submission for a student
I know have added the same submission for two different students
I select it in the list, nothing changes (as two identical essays were added this is expected)

(user switches between essay content and original formatting)
Hm, its the same text but displayed differently. Ones seems to be without formatting
I'm not sure if there is another difference and what it is good for
Its unused for me that I have to click on the tab that is marked (refers to the rending of the tabs, confuses active and inactive tabs)

O: Please create a rubric now for evaluation

(opens rubric select dialog)
I have only two options...cancel means to abort something...new to create. I choose new.
(new rubric creation wizards open)
I enter some title
Ok, I need to activate something
(users selects a plugin and activates it)
I need to close this to continue...or...(user closes screen)
I' need to wait...
O: Why? You can continue to work?

But here it seems to be running (points to second monitor)

O: Oh sorry, you can ignore that. This is the IDE and has no meaning in the program we test. Please continue with the task.

I click on finish.
Hm...what can I do? What did I have selected?
(user sees rubric edit view, but no criterion. User had activated a plugin but didn't select the criterion in the last step of the rubric creation wizard).
With that button I can go back

I think I need to add something
(open add examples dialog and adds a positive example)

Here it seems I can set the weight. But if I should to a weighting of the criteria I need more than one.

I remember I had chosen one but I don't see it.
I'll try to add one more

I think I forgot this step before. To activate it. I do it now. Ok, now finish.
(repeats the steps done before to add a criterion. activates anther plugin but again does not select it in the list to actually add it)
So, were is my criterion?
O: I'll help you here as you seem to have some trouble. Open up the dialog to add a criterion again. But before you finish make sure to select it in the list.
Ok, I'll try (repeats steps and successfully adds the criterion now)
Oh now I see something displayed here.
I'll add the second one too

I assume I have to do a weighting now
What do the numbers mean? The higher the more important the criterion is?
I've done a weighting of the two criteria.

Edit the criterion? I'll see
(selects spelling criterion and opens the edit dialog. starts reading the instructions)
I can define up to 5 levels. The higher the level the better
I specify the maximum amount of allowed errors for each level
I enter 6, 3 and 0 for the three levels. I'm not sure what this list is, I close it now (list is the wrong words list)

And the same I should do here (means the second criterion, grammar)
Ok, I specify the error counts again

I can close it now (goes back to the main screen, successfully selects rubric)

Do I need to start the process now somehow?

O: Is there everything you need done?

There are the essays and we have the rubric. I think so.
Ah, here is evaluate
It should have evaluated the selected essay now

The result means to me that grammar was found to be good but spelling to be poor. It's from the automatic evaluation

(opens full rubric view)
Here I see a more detailed result

I'd like to see the errors in the text. I see a list below but I want them marked in the text.
(starts searching...takes a while to find the highlight button)
Maybe in windows? No...but I only have the buttons at the top...
Now the spelling errors are highlighted

I can search for something (opens find dialog)
It found the essay of the student
There is an X but nothing happens. I'm used that this closes the window. I have to click on the quit button, that is strange

O: What do you do if you are not satisfied with the automatic results?

I'd expect to be able to mark a part in the text and enter something. This should change the achieved level

O: This is not possible in exactly this way yet. But there are ways to change the achieved level. Please correct an automatic rating now.

Then I expect that I can change the level.
(Clicks on rubric criterion, selects a level)
I successfully changed the result. And the level. It now says manual rating
(opens criterion again)
Ok, the dashed one is the automatic rating and the manual one is solid. The manual changed the result. So it overrides the automatic rating. That's good

O: There is a second way to alter the results
There are some words marked as wrong which are correct. That should be editable.

You said I can't do it in the text. Maybe in the list here?
(user selects a word)
It highlights. I think it should be possible this way
(tries and does right click after a while)
That should do it
Ok, the error count is updated
And I would like that it remembers my correction. I don't want to do it again.

The symbols in the essay list changed
I wonder what this means
Ah, it says they need reevaluation. So it remembers that I marked the word as correct in the other essay
When I select another essay it automatically starts again to evaluate

O: Let me change the assignment and used rubric. Then we will continue with the test. You have added an example essay before. But in this rubric there is none. Please add one and see what can be done with it.

Ok. I need to edit the rubric to do it.
It tells me it will delete the results. That's clear, when I change the rubric the results get invalid
Now there is an example
I'm not sure what I can do now with it.
I'd expect it is used to configure the criteria. But I don't see how

O: Try to edit the spelling criterion

There is a list of wrong words. Maybe there are coming from the examples. But I'm not sure about this. But I can mark words as correct here. I expect they will be treated as correct later in evaluating the essays.

O: Thank you for participating in the test. Do you have any remarks now?

I think its very good that the system remembers that the essays need to be reevaluated. But what I miss is to directly enter corrections in the text.And some things don't work like I'm used to, especially the X to close a window.

I. Plugin and Criterion Interfaces

Interface RubricAnalyserPlugin:

```
public interface RubricAnalyserPlugin {

  /*  @return An String ID for the plugin. ID must be unique
   */
  public String getPluginID();

  /*  @return the number of criteria the plugin provides
   */
  public int getCriterionCount();

  /* @parameter data stored serialized instance
   * @return deserialized Criterion instance with all transient members
   *         correctly initialized
   */
  public Criterion deserializeCriterion(byte[] data) throws PluginException,
                    InvalidClassException;

  /* @parameter index the requested index of the criterion. Valid range
   *                  0 to getCriterionCount()-1
   * @return Criterion instance
   */
  public Criterion getCriterion(int index);

  /* will be called when plugin is loaded so it can set up internal states
   * ATTENTION: also called when the plugin settings change to verify the
   * settings are ok
   * @return: true if plugin is ready for use. false if plugin requires
   *          more configuration
   */
  public boolean initalize(PluginMetaData meta) throws Exception;
}
```

Interface RubricAnalyserPluginGUI:

```
public interface RubricAnalyserPluginGUI {

  /* will be called when plugin is loaded so it can prepare internal
   * states for its GUI
   */
  public void initalizeGUI();

  /* @return true if the plugin has a configuration screen
   */
  public boolean providesPluginConfigurationScreen();


  /* Settings screen for the plugin
   * Setting should not be automatically saved (instead
   * saveConfiugrationScreenSettings will be called)
   * @return JComponent providing the plugin options to be manipulated by
   *         the user.
   *         It's recommended that the returned Jcomponent is a JPanel or
   *         an similar acting component
   */
  public JComponent getPluginConfigurationScreen();

  /* Save settings made in the configuration screen
   */
  public void saveConfigurationScreenSettings();

  /* Reset the internal plugin configuration to default values
   */
  public void resetConfigurationToDefaultValues();

  /* @parameter size: must support the sizes: 8, 12, 24 pixel
   * @return Icon, size x size (square) used as logo for the plugin.
   *         Should return any size which is a multiple of 4.
   *         if a size is not supported null may be returned.
   */
  public Icon getLogoIcon(int size);

}
```

Abstract Class Criterion:

```
public abstract class Criterion implements Serializable {

  public static final int NO_LEVEL_REACHED = -1; // Constant
  public static final int REEVALUATE = -2;       // Constant

  protected Criterion(String title, PluginMetaData meta_data)
  protected Criterion(String title, PluginMetaData meta_data, Rubric r)
  protected Criterion(String title, PluginMetaData meta_data, Rubric r, int weight)

  final public int getWeight()
  final public void setWeight(int weight)
  final public String getTitle()
  final public int getAutomaticLevelRating()
  final public PluginMetaData getPluginMetaData()
  final public void setPluginMetaData(PluginMetaData meta)

  /* indicates if the criterion has been modified since the last call
   * returns true only for changes which result that previous evaluations results
   * will be invalid with the new criterion parameters
   * status is set back to unchanged after the call
   * @return true if it has been modified since last call
   */
  final public boolean hasChanged()

  /* @return the index of the criterion within the rubric
   */
  final public int getCriterionIndex()

  /* @return short description of the criterion
   */
  final public String getDescription()

  /* called after a successful call to evaluate
   * return an appropriate message to the latest done automatic evaluation
   * @return null or a short one line message
   */
  abstract public String getRubricAutomaticLevelMessage();

  /* @return instance of the Plugin which provides this criterion
   */
  final public RubricAnalyserPlugin getPlugin()

  /* @param needed when deserialzed to set transient members correctly
   */
  final public void setRubric(Rubric r)

  /*@qreturn return the rubric the criterion belongs to
   */
  final public Rubric getRubric()
```

```
  /* returns an ID for this criterion. ID must be unique within the
   * plugin providing the criterion
   * @return ID for this criterion
   */
  abstract public String getCriterionID();

  /* @return a list of subcriteria if there are any. return null
   *   if no subcriteria exist
   */
  abstract public List<SubCriterion> getSubCriteria();

  /* @return success
   */
  abstract public boolean evaluate(DocumentData data) throws PluginException;

  /* @param data has been evaluated before, therefore data contains old markup.
   *        needs to be removed and updated
   * @return success
   */
  abstract public boolean reevaluate(DocumentData data) throws PluginException;

  /* @param index, valid range 0 - getLevelCount()
   * @return true the level specified by the index.
   */

  abstract public Level getLevel(int index);

  /* @return true if the criterion is set up correctly and ready to be evaluated
   */
  abstract public boolean isValid();

  /* @return the number of levels
   */
  abstract public int getLevelCount();

  /* maybe change this later..needed for persistent storage..
   * to deserialize use deserializeCriterion method of the implementing plugin!
   */
  abstract public byte[] serialize();

  /* called after a criterion was deserialised and the rubric was set
   * (rubric is a transient member therefore it is not possible to initialise
   * listerners before it was set).
   */
  abstract public void initRubricListeners();
}
```

Interface CriterionGUI:

```
public interface CriterionGUI {

  /* @return true if the criterion has a configuraition screen
   */
  public boolean providesCriterionConfigurationScreen();

  /* The returned JComponent will be displayed by the main framework
   * (It's recommended that the Jcomponent should be a JPanel or an similar acting comp
   *
   * @param close_action: predefined action which must be used to close the
   * configuration screen. Must be attached to e.g. a button in the returned JComponent
   * @return the JComponent providing the configuration options
   */
  public JComponent getCriterionConfigurationScreen(CloseViewAction close_action);

  /* @param document for which the highlighters are requested
   * @return List of Highligter Action, may be null if none are supported
   */

  abstract public List<ActionHighlight> getHighlightActions(DocumentData document);

  /* @param document document for which analysis details should be shown. property chan
   * should be registered on document to update the dispaly when data changes
   * @param width, height: maximum panel dimensions
   * @return returns a jpanel contianing details for the current doucment analysis. pan
   * to hightlight details in document dynamically
   * may return null if not supported
   */

  abstract public CriterionGUIDetailPanel getCriterionDetailsPanel(DocumentData documen

  /* @parameter size: must support the sizes: 8, 12, 24 pixel
   * @return Icon, size x size (square) used as logo for the plugin.
   *         Should return any size which is a multiple of 4.
   *         if a size is not supported null may be returned.
   */
  abstract public Icon getLogoIcon(int size);

}
```

J. Licenses

List of licenses of used software libraries and databases. Full license texts are included on the accompanying CD-ROM.

- MiGLayout: BSD and GPL, http://www.miglayout.com
- IDW: GPL and commercial, http://www.infonode.net/index.html?idw
- SwingX: LGPL 2.1, http://java.net/projects/swingx
- Flamingo: BSD, http://java.net/projects/flamingo
- Cobra toolkit: LGPL, http://lobobrowser.org/cobra.jsp. The toolkit includes the dual Mozilla Public License (MPL) 1.1/GPL 2.0 licensed Rhino 1.6R5 Javascript engine released by the Mozilla Foundation.
- Jxlayer: BSD, http://java.net/projects/jxlayer
- Wizard: CDDL-1.0 ,http://java.net/projects/wizard
- Substance: BSD,http://java.net/projects/substance
- Substance-swingx: LGPL 2.1 ,http://java.net/projects/substance-swingx
- Stanford POSTagger: dual licensed under the GPL for research and open source software and under a commercial license for proprietary software, http://nlp.stanford.edu/software/tagger.shtml
- Jazzy: LGPL V. 2.1, http://jazzy.sourceforge.net
- LanguageTool: LGPL, http://www.languagetool.org
- WordNet: custom license allowing commercial usage, http://wordnet.princeton.edu/wordnet/license
- RiTA.WordNet: GPL, http://rednoise.org/rita/wordnet/documentation
- MorphAdorner: custom NCSA style license, http://morphadorner.northwestern.edu/morphadorner/licenses
- Apache Derby: Apache 2.0 license, http://db.apache.org/derby
- Connector/J: GPL 2.0 and a commercial license, http://dev.mysql.com/downloads/connector/j/5.1.html
- Snowball (Porter2): BSD. http://snowball.tartarus.org
- JBusyComponent: LGPL 2.1, http://code.google.com/p/jbusycomponent
- MimeUtil: Apache 2.0, http://sourceforge.net/projects/mime-util/
- JPF; LGPL 2.1, http://jpf.sourceforge.net