

An Automatic Grading System based on Dynamic Corpora

Djamal Bennouar

Bouira University, Algeria
dbennouar@gmail.com

Abstract:

Assessment is a key component of the teaching and learning process. Assessing a student's answer to an open ended question, even if it is a short answer question, is a difficult and time-consuming activity. Current Automatic Grading Systems (AGS) achieve their works using static corpora. Building efficient corpora for a course is actually a challenge. The underlying subjectivity in grading short answers may have a serious impact in the quality of a corpus. Specific course context defined by a teacher and the time dependent grading strategy may make very difficult the construction of traditional course corpora. This paper presents an AGS for short answer based on dynamically built an up to date corpora. The corpora contain two kind of corpus: a corpus related to the reference answer and a corpus related to the student answer. Each corpus is automatically generated by applying a set of semantic and syntactic teacher indications to the reference and the student answer. The teacher indications are introduced by a teacher in a process of predicting possible student answer. The grading process of the proposed AGS tries to find the most similar answers in the two corpus in order to determine the most correct grade for a student answer.

Key words: Computer Aided Assessment, Automatic Grading System, short answer, corpus, answer predicting, text similarity

1. Introduction and Motivation

Assessment is a key component of the teaching and learning process. Indeed, it is asserted that the effective use of assessment practices can improve the quality of teaching and learning [Iahad, 2004; Biggs, 2007]. Two main kind of assessment technique are commonly used: assessment based on selection-type question (also called objective-type questions) and assessment based on open-ended questions. Selection-type questions which include multiple-choice questions, true/false questions, matching questions, etc., suggest a selection from predefined choices whereas open-ended questions require the student to express himself/herself in composing the answer in his/her own words and style. Even though writing and the textual form are the most common, there are other ways to present data and ideas for open-ended questions, like diagrams, schemas. In the remainder of this paper, we will refer to open-ended questions with the text form as essay-type questions that require free-text

There are two forms of the essay-type question, namely short-answer question and essay question (Heinrich, 2006; Ziai et al, 2012). Short answers have a limited length and are supposed to target defined problems. They are usually supposed to contain only a few facts that answer only one question (Ziai et al, 2012). Essay questions, in contrast, are not limited in length and students are given freedom of response in free text with the only help of their own experience and knowledge. Thus, research around short-answer assessment differs from essay scoring.

It is usually argued that Short answer Questions are typically used for assessing knowledge (Perez 2007). Actually this view on short answer cannot be applied to any kind of learning process. This view may be correct in the context of some social sciences learning process where a long answer is needed to discover some student's skills and capabilities. In some technological fields like computer science and electronics, an answer is sometimes just a value. However to produce such kind of very short answer, a student has to stir his brain in order to show his capabilities and skills. As an example, in computer architecture, programming language, or computer algorithms, finding a group of missed instructions in a program function requires many capabilities and skills, not only the knowledge about the language syntax. The student has to: discover the problem addressed by the incomplete function, understand the objectives of the function, discover the function logic, decides what are the instructions needed to produce a complete function and finally test the function.

In most Algerian Universities, except in medicine discipline, where the main assessment technique is based on selection type question, teachers in all other discipline uses open ended question to assess their student. In technological discipline such as computer science, electronics, civil engineering, short-answer question seems to be the most used, while in social science such as law, psychology, essay-type question represent the most approach. The reason given by teacher using open ended question is the fact that selection type question are not adapted to assess student quality for the courses they are conducting. They argue that the content of a written answer may contain many elements which allow an efficient measure of the student's knowledge of the topic.

Writing answer to an open ended question actually requires more thought than answer to selection-type question, since the students must construct their own coherent answers and justifications (Perez 2007).

Actually the two arguments cope with the result of many research activities. Sigel (1999) reports cases in which students with high scores in Multiple Choice Questions, have shown deep underlying misconceptions when interviewed by a teacher. According to many research (Birenbaum et al., 1992; Mcgrath, 2003; Mitchell et al., 2003; Palmer and Richardson, 2003) open ended questions represent the efficient approach if the teacher target to assess the student ability to synthesize and analyze information; to find new connections between ideas and to explain their significance.

In most Algerian Universities, assessing a student's answer to an open ended question, even if it is a short answer question, is a difficult and time-consuming activity. In addition to the great number of student enrolled in a course, there are three other main reasons which make teachers facing a hostile environment for assessing their students:

- The poor level language, either in Arabic or French of the majority of student. Written text is usually full of misspelling and wrong sentences construction. Homophone represents one of the main sources of errors in text writing.
- The great number of students who ask for the consultation phase. In Algerian Universities, the assessment of a student answers goes through two phases: In first phase teacher evaluates student's answers and grades them. In a second phase students are invited to consult their answer by comparing them to the correct answer provided by the teacher. This second phase is time consuming and difficult to achieve. Often, for each student the teacher has to enter a hard dialog with the student. Usually, the teacher rereads with the student the answer, takes into account the student's argument, does his best to convince student that some part in the answer are errors and finally, if necessary, adjust the student's grade.
- The unclear hand written answers: many alphabetical letters, digit numbers or symbols are written intentionally or not in a similar manner leading to turbulent and stormy dialogue between teacher and student in the consultation phase.

In order to enhance the quality of the learning process and the global student evaluation process and to highly reduce assessment time and its difficulties, most Algerian Universities were provided with an e-learning environment as the result of a government initiative. An eLearning environment is usually represented by at least one classroom provided with a number of desktop computers connected to a server machine hosting an Learning Management System (LMS). Unfortunately, such environment seems to be rarely used in the student's assessment process. As an example, Bennouar (2013) reported that in the Blida University, which count more than one thousand teachers, only less than a dozen of teachers use the LMS server only as a repository for their courses and at most two teachers use the LMS server through internet to assess student using selection-type question. From a scientific and technical point of view this situation is due to the main following reasons according to (Bennouar 2013)

- Like some skeptical researchers (Perez 2007), many teachers are not convinced that Automatic Grading Systems (AGS) have the capacity to efficiently grade free text answer for their courses, even for short answers.
- Automatic Grading Tools for short answer, included in the provided LMS platforms, are either not efficient to support assessment (Bennouar 2013) in an hostile environment or are not simple to use even by computer science teachers

According to (Bennouar 2013), providing an efficient AGS for short answer represents a key element to get teachers interested by eLearning environment in their learning process. An efficient AGS for short answer has to be provided with:

- The capacity to deal efficiently with a hostile environment characterized by answers having a poor language level, misspelling, missing word, not required words, homophone etc.
- The capacity to let teacher indicates what is important and not important in an expected student's answer
- The capacity to apply teacher specific penalty to an answer containing partial errors
- The ability to be adapted to any kind of course and language

The core functionality of an AGS for short answer is represented by a text similarity process which compares a student's answer to a set of predefined answers organized in the context of corpora, and assigns, according to the comparison results, a qualification value to the student's answer which may be a grade or an appreciation. Corpora used in current approaches are usually constructed manually, deals with a specific kind of

courses, answers and language are very hard to update. This paper presents an approach which provides dynamic and up to date corpora for automatic grading of short answer. The automatic building process of such corpora is mainly based on the teacher predicting activity of probable student answer. The key concepts of such automatic building process is its capacity to be tuned by teachers to deal with the kind of course, the objective of the short answer question, the student's knowledge and skill and the student's language level. Building efficient corpora means that in one side, student's answer receives the correct grade, and in another side, the number of student claiming for a manual review of their answers has to be very limited.

2. Related works:

Short answer assessment tools compare a student's answer to a set of predefined answers, called model answers, and assign, according to the comparison results, a qualification value to the student's answer which may be a grade or an appreciation. The comparison process which represents the core of such approaches is highly related to the task of text similarity (Mohler 2009) which is essentially the problem of detecting and comparing the features of two texts. Text similarity depends highly on the presence of a data set represented usually a corpus. The efficiency of a corpus is vital for the success of an automatic grading process.

To our knowledge, current corpora are manually built based on student answers gathered from previous tests. The only kinds of corpus which may be seen as an up to date automatically generated corpus are those related to tools using regular expression to specify a reference answer. The main goal of regular expression is to compress in a single expression a great number of alternative and equivalent answers. Moodle Regex (Moodle 2015) and WebLas (Bachman 2002) are kind of such tools. In Moodle Regex, teacher has to specify manually the answer as a regular expression.

The main drawbacks of such kind of tools are:

- The difficulty to understand regular expression concepts and the huge difficulty to operate them. Even computer science teacher find difficulties with regular expression specification.
- All the alternative answers which correspond to a regular expression receive the same grade.
- Some kind of errors, like homophones, are very difficult to represent using regular expression

In WebLas (Bachman 2002), a limited regular expression is created automatically for each model answer after defining, under the control of the teacher, after determining the possible alternatives for each important element using WorldNet. Actually the regular expression of WebLas has the merit to ease the creation of multiple similar model answers instead of letting the teacher using either the regular notation himself. According to (Bachman 2002), WebLas regular expression are very simple since they are built using only the alternative operator and seems to be oriented to deal with a specific kind of assessment (English language ability). In addition, according to (Ziaiet al, 2012) the WebLas miss an evaluation study based on data.

Manual built corpus needs to be manually updated each time to support new aspect in a course and actual approach followed in the teaching process. The teacher specific point of view of the assessment process and the various kind of question used in an examination (enumeration, justification of a choice, definition, description etc..) are other great aspects which has to be taken into account when updating a corpus. The task of updating a corpus is actually very hard to achieve.

The static grades assigned by experts to each answer in a corpus may be also a source for the reduced efficiency of a corpus. None of used corpora took into account the fact that an answer is evaluated according to a grading strategy which may evolve during the teaching process. The grading strategy is decided by the teacher and may be specific for each answer in an online exam. The teacher decides for each question what is important and not important, what is the penalty to apply for a partial answer, what are the mistakes which have no impact on an answer and what are the constructions which destroy all the answer even if the answer contains correct part. The major parts of the recent research reuse existing manual built corpora and focus only in enhancing the similarity process.

Based on their corpus and the used text similarity, many current academic and industrial AGS reported a high level of satisfaction. However the evaluation were actually obtained in a favorable environment characterized by two main aspects:

- The presence of corpora with a great number of answers related to a specific kind of question, a specific language and a specific part of course, graded by some experts.
- The answers provided in the corpus and by the students are usually correctly written and in some rare case, a student's answer may contain simple spelling errors.

Actually such favorable environment does not match the Algerian University reality and the reality in most third world universities. The actual environment is rather hostile for automatic grading of answers. As an example, the spelling errors and homophone based errors represent the rule in student's answer and even some

answers provided by a teacher may also contain errors. Current corpus doesn't deal seriously with such incorrect answer.

The lack for corpora ready to be operated by any grading approach or tools is another characteristic of the hostile environment. To my knowledge, in Algerian Universities, where automatic grading seems to be very rarely used, there are no corpora or knowledge bases oriented to assess student in any course. In addition, building efficient corpora or knowledge base for a course is actually a challenge for most Algerian Universities and most third world universities. The underlying subjectivity in grading short answers may have a serious impact in the quality of a corpus (Mohler 2009). In the dataset used by (Mohler 2009), some grades reported by experts differed in some situation by more than 4 points on a five point scale.

In addition to the underlying subjectivity, specific course context defined by a teacher during a teaching process and the time dependent grading strategy may make very difficult the construction of course corpora. The course context information recognized by students as part of the course learning process, are either not reported at all or reported using just a referencing technique, like saying "according to the programming style defined in course's section x". With such hidden information, it will be very difficult for student not enrolled in a course with a particular teacher to provide a correct answer.

The time dependent grading strategy is due to the fact that in a university context, the student evaluation is not a static process but rather a continuous process which highly depend on time spent by a student following the course. In early steps of a teaching process, the evaluation of an answer may be tolerant for some error. This tolerance will be reduced in next steps. In a final exam this tolerance may not be accepted at all and some errors, accepted in early stages, may be fatal for an answer, even if a part of an answer is correct. This change in the evaluation strategy seems to have not been considered in miscellaneous AG approaches and tools and may represent a challenge for building efficient corpus.

In addition, current corpora suffer from a number of lacks, mainly the answer dynamic content and multiple short answers.

Answer dynamic content: in some answers, the student has to provide his proper words or his proper values. This situation may be found when a student has to choose a variable name or a value to initialize a variable when writing a programming language expression, a subject name when composing a small sentence etc... Such words and values become accessible when the student submits his answer to the automatic grading system. Hence a model answer may contain two kinds of contents: a static content, known when the model answer is defined and a dynamic content which is decided during the process of elaborating an answer by a student. When a number of students provide their answers, the provided content for a dynamic part may differ from one student to another. To support dynamic contents, the specification of a model answer has to be provided with mechanisms which let the AGS differentiate between dynamic and static contents. To our knowledge none of the current corpora, approaches and tools deals with such kind of answer.

Multiple short answers: In some situation an answer has to contain a list of short answers separated by a predefined mechanism such as a comma or a new line. Multiple short answers may be elaborated when reporting a list of characteristics, the steps of a process, the words needed to fill part of a text, a small part of a computer program etc.. In some cases, the reported short answers are independent and may be reported in any order. In some other situations, the short answers have to be reported in a specific order. Dynamic content may exist in multiple short answers and may be shared between two or more short answers belonging to the same multiple short answer. Multiple short answers technique has to take into account many challenges such as the impact of additional non awaited short answer and multiple report of a same short answer.

3. The proposed Approach

3.1. System overview

Figure 1 reports the global architecture of an AGS based on a dynamic and up to date corpora for short answer grading process. This architecture shows that the grading process goes through two important phases: the preparation phase and the grading phase. The preparation phase, which has to be incorporated as a first step in an AGS, achieves the most important intelligent works in the system in order to make efficient the grading phase which is specific for an AGS. The result of the preparation phase is represented by two corpora: the Grading Engine Reference Answer (GERA) and Grading Engine Student Answer (GESA). The grading phase is mainly based on text similarity approaches and may be specific for each AGS.

In the process of preparing a question, a teacher submits to the preparation phases a set of Reference Answers (RA) and a set of indications. Based on the teacher indications, the preparation phase produces for the grading phase a rich ordered set of fully or partially correct answers with their corresponding grade. This set is called the Grading Engine Reference Answer (GERA). A similar process is also applied to the student answer

and a set of student answer called Grading Engine Student Answer (GESA) is then generated for the grading phase.

The grading process try to find among the GERA answers and the GESA answers the most two similar answers, one from GERA and the other from GESA. According to two predefined similarity thresholds, two kinds of decisions may be taken by the grading phase:

- Associate to the student answer the grade of the corresponding GERA answer, if the computed similarity is greater or equal the Automatic Grading Threshold Similarity Factor (AGTSF).
- Report the answer as an answer which has to be manually reviewed by the teacher if the similarity factor is greater or equal the Manual Grading Threshold Similarity Factor(MGTSF).

3.2. Specification of reference answers and teacher indications:

Once the reference answers set are completely defined, the teacher associates a number of indications with each answer. An indication is always associated with a penalty factor. The application of an indication to a reference answer produces other reference answers tagged with the indication penalty factor. Through the indications, the teacher tries to predict the various forms of student answers which may be considered as fully or partially correct. In fact, through this prediction task, which may be time consuming, the teacher tries to prepare the best environment for the AGS in order to highly reduce the number of automatic grading decision reject and the number of decision asking for a manual grading.

The specification of reference answers and their associated indications is achieved using an XML based language called the Short Answer Specification Language (SASL) [Djelattou 2015]. The teacher indications are represented by specific XML tags and regular expression meta-characters. The main objective of the specification of reference answers accompanied with a number of indications is to efficiently guide the preparation and the grading phases.

The use of XML tags and regular expression, even by computer science teachers is not easy at all. Like other XML based system, instead of the direct use of XML notation and complex regular expression, the teachers are provided in AGS with two simple facilities to specify their indications: The Common SASL Annotation and the Teacher Interface.

The SASF Common Annotations (Table 1) were introduced to ease the specification of various simple indications directly in the reference answer text. Short answer dynamic content and key words are examples of such simple teacher indications (Figure 2).

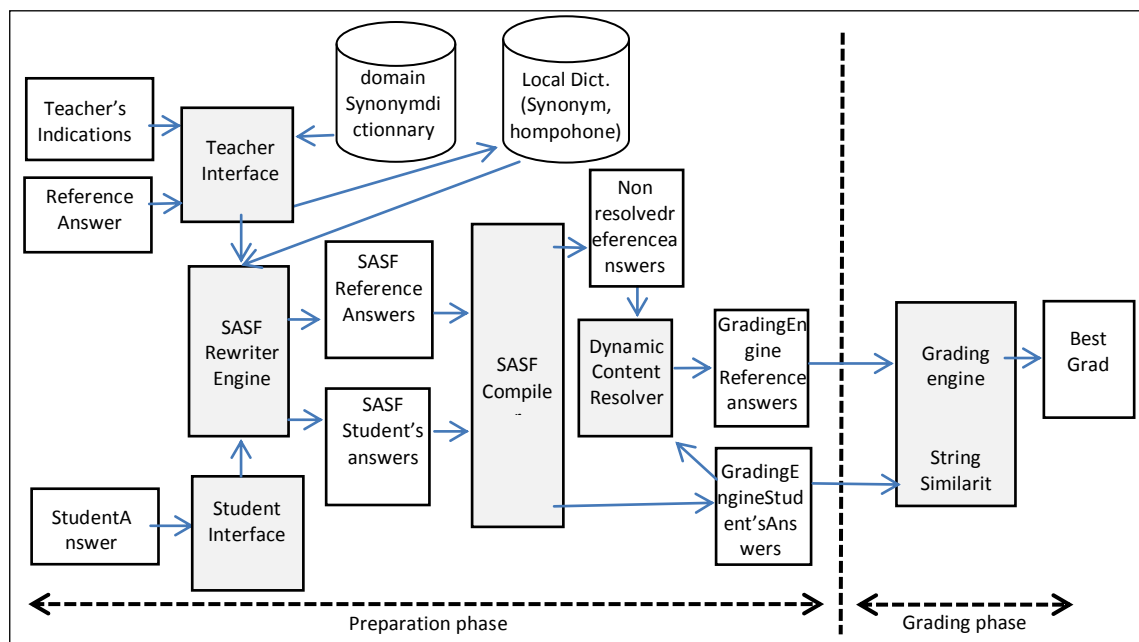


Figure 1: AGS Architecture

The question:(Original question was in French language)

According the java source code writing style, write only the first line of the constructor used to initialize the last name and the firstname of an object instantiated by the following Personne class.

```

public class Personne {
    String matricule, nom, prenom ;
}

```

Note that the constructor may be used by any other class from any other package to build object

A reference answer :

```

$key=public $key=Personne(String $alpha_id1, String $alpha_id2) $key=

```

Figure 2: An example of answer written using dynamic content and keywords

SASF Common Annotations	Role	XML tag
\$list{} \$list() \$list'''		<LIST /LIST>
\$alpha, \$alpha_id	Alphabetic name	<DYNC /DYNC>
\$alphanum, \$alphanum_id	AlphaNumeric name	<DYNC /DYNC>
\$integer	Integer value	<DYNC /DYNC>
\$key: \$key:{} \$key:() \$key:'''	Keyword designation	<KEYW /KEYW>
\$key: {theKeyWord, keychar=aValue, maxcar=aValue}	Number of important character	<KEYW /KEYW>
\$syn: \$syn:{} \$syn:() \$syn:'''	Enable synonymy	<SYNONYM /SYNONYM>

Table 1: sample SASF common annotations

Teacher indications which are difficult to directly report in the reference answer text are specified using the Teacher Interface. Among such indications we can find the maximum grade associated with each answer, the Automatic and Manual Grading Threshold Similarity Factors, the penalty factors associated with each indication, the necessity to ignore or consider key word order, case sensitivity, stop words, character accent, homophone, synonymy etc... Teacher indications are organized in classes, each one managed by a sub controller in the Teacher Interface. Table 2 reports the main indications classes supported in the current version of SASL and table 3 shows the definition of more indications to be applied for the answer shown in figure 2.

Indication Classes Controller	Role
Grading controller	Define maximum grade and similarity thresholds, select similarity tools and define execution order
Stop word subcontroller:	Stop word definition and consideration (ignore, replace)
Character Controller	case sensitivity, char repetition, char replacement, separator definition, character accent
Keywords controller	Point out important word in an answer (Key words), ignore/consider Keyword order, accept missing keyword,
Synonym subcontroller	Build and activate the local synonym dictionary, select words for which synonym have to be considered
Homophone controller	Define which homophone are considered as similar to the one proposed in the teacher model
Dynamic part controller	define words which cannot be used in dynamic part keywords, enable style controller (unauthorized words, used character, number of precision digit), out of range control (interval definition, accepted error in precision)

Table 2: A sample of teacher Interface sub controllers and SASF indications classes

Some teacher indications, like the consideration of synonymy or homophony, takes the preparation phase into a process of generating a local dictionary under the full control of the teacher. More global resource, like a domain synonym dictionary may be used by the preparation phase to ease the process of creating the answers local dictionary

SASL indication	Default status and penalty in Teacher Interface		Teacher status selection and penalty	
Preprocess: Pack text	ON	0%	ON	0%
Extra Char / Word (Student answer only)	ON	Shared word 30%, char 10%	ON	word 20% char 5%
Keyword Order	ON	100%	ON	100%
Keyword miss (public, {})	ON, all	Shared (100%):public = 33%, Personne = 33%, { = 33%}	ON, select	Not shared: Public=10% { = 5% Personne 100%
Dynamic Content style	ON, all	0%	On, all, not cumulative	5%
Key char in Keywords	OFF, all	0%	ON, select: Public (S3, M5) Personne (S4, M9)	Public : 5%, Personne: 5%
Character Repetition	OFF	0%	ON	5%
Case sensitivity:	ON, all	0%	OFF, all	10%

Table 3: Example of indications for the question of figure 3, specified using Teacher Interface

3.3. The SASF Compiler

The student answers, the reference answers and the teacher indications specified using SASL Common Annotations or the Teacher Interface, are translated by the Answer Rewriting Engine (ARE) to a full SASL description. An SASF answer description contains the original text answer and the miscellaneous indications which have to be applied in the process of building the Grading Engine reference and student answers.

The specified indications are not applied systematically to the reference answers and to the student answers. Some of them, like “accept missing keyword” (penalty factor is less than 100% for a non-mandatory keyword) target teacher answers only. Other indications target either teacher or student answers. This is the case of “pack answer text” specified in the preprocess indication class, stop word consideration (ignore, replace) and homophony equivalence. A third kind of indications, like extra lines, extra words or extra characters indications, target only the student answers.

Id	Indication Generated answers	Penalty
0	public Personne (String \$alpha-id1,String \$alpha-id2){	0
1	public Personne(String \$alpha-id1,String \$alpha-id2){	0
2	public Personne(String \$alpha-id1,String \$alpha-id2)	5
5	public Personne(String \$alpha-id1,String \$alpha-id2){	5
6	Personne(String \$alpha-id1,String \$alpha-id2){	10
11	pub Personne(String \$alpha-id1,String \$alpha-id2){	10
24	personne(string \$alpha-id1,string \$alpha-id2){	20
27	public per(string \$alpha-id1,string \$alpha-id2)	20
36	persone(string \$alpha-id1,string \$alpha-id2)	30

Table 4: A Sample of Compiled non resolved reference answers

The main task of the SASF compiler is the production of an ordered set of reference answer called the Grading Engine Reference Answer (GERA) (Table 4) and an ordered set of student answers called the Grading Engine Student Answers (GESA) (Table 5). To achieve this task, the SASL locates the next indication to perform using the indication class priority. Once located, the indication is applied to the original answer and also to answers produced in the previous step. The preprocess class indication has the highest priority. When applied without any penalty, the produced answer replaces the original one in the grading phases. Due to the successive application of indications and their penalties, the first produced answer is associated with the maximum grade and the last produced is associated with the smallest grade.

When a reference answer contains a dynamic content, the SASL compiler produces a set of non resolved reference answers (Figure 4) which needs to be submitted to a resolution process in order to produce the Grading Engine Reference Answer. The dynamic content resolution process, called the Answer Dynamic Content Resolver, is activated when a student answer is compiled. The Answer Dynamic Content Resolver try to locate in the student answer the corresponding content. Once located, the resolver verifies if the miscellaneous dynamic content properties are present or missed in the located content, and if necessary, applies the corresponding penalty. In case of the example reported by table 4 and table 5, the string **Nom** and the string **prenom** will replace the dynamic content **\$alpha-id1** and **alpha-id2**. The Answer Dynamic Content Resolver has to insure that the string **Nom** is different from the string **prenom** and all of them were written using only lower case alphabetic characters.

N°	Indication generated student answers	Penalty
0	class publique personne (string Nom, string prenom);	0
1	class publique personne(string Nom,string prenom);	0
5	publique persone(string Nom,string prenom)	5
6	pub pers(string Nom, string prenom)	10
11	publique persone(string nom,string prenom)	15
13	pub persone(string nom, string prenom)	20
GESA specific penalties: 30% (Added to basic GERA answer penalties)		
Style= 5% : personne, Nom		
Extra Word = 20% : class		
Extra Char = 5% : ;		

Table 5: Some answers from the Grading Engine Student Answers

Conclusion and future work

This paper has shown the importance of automatic building of up to date corpus for a short answer automatic grading process. The main idea in the approach presented in this paper is to provide the teachers with facilities which help them to easily predict the various possible student answers. The prediction is achieved using a number of indications which are then applied to the original reference answers to produce a rich corpus containing answer with their associated grade.

This approach is currently under implementation in the context of the Moodle LMS platform. The first experiments of some part of the system (blank, accent, keyword, partial keyword) using only the n-gram similarity metrics in the grading phase, has shown that this approach is very efficient to assess student if some

computer science courses like Java language, Object Oriented Programming, Component Oriented programming and Computer Networks. The teacher indications have made the corpus independent from the kind of question and robust regarding the various kind of frequent language error, mainly accent, missing characters and homophone. Future work will focus on completing the miscellaneous aspect of the SASF language and its adaptation to the Arabic language.

The main planned drawback (not yet experienced in an actual exam) of this approach is the huge number of generated answer when a teacher uses deeper indications. As an example, in an answer specified in the French language, if a teacher penalizes each missing accent for a short answer containing five accents, the systems generate all possible alternatives and apply for each alternative the corresponding penalty. A corpus with huge number of answers may be a source for increasing the response time of the system in an actual course examination.

References

- Bachman, L., Carr, N., Kamei, G., Kim, M., Pan, M., Salvador, C., Sawaki, Y., 2002. A reliable approach to automatic assessment of short answer free responses, Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002), pages 1–4.
- Bennouar, D., 2013. Challenges of e-Test in a University context, the 1st Elearning Spring School, SaadDahlab University, Algeria, 27-30 April 2013 (in french)
- Biggs, J., Tang, C., 2007. Teaching for Quality Learning at University Third Edition (McGraw-Hill.). Open University Press
- Birenbaum, M., Tatsuoaka, K., Gutvirtz, Y., 1992. Effects of response format on diagnostic assessment of scholastic achievement. Applied psychological measurement, 16(4), 1992.
- Djelattou, R., 2015. A Short Answer Specification Language for an Automatic Grading System, Magister thesis, Computer Science Department, the SaadDahlab University, Blida, Algeria, Feb 2015.
- Heinrich, E., 2006. Addressing Efficiency and Quality of Marking in Essay Assessment with E-learning Support. *Journal of Distance Learning*, 10(1), 15–25.
- Iahad, N., Dafoulas, G. A., Macaulay, L. A., 2004. Evaluation of Online Assessment : The Role of Feedback in Learner-Centered e-Learning Learner-Centred Paradigm. In *37th Hawaii International Conference on System Sciences*, pp. 1–10.
- Mcgrath., P., 2003. Assessing Students: Computer Simulation vs MCQs. In Proceedings of the 7th Computer Assisted Assessment Conference.
- Mitchell, T., Aldridge, N., Williamson, W., Broomhead, P., 2003. Computer based testing of medial knowledge. In Proceedings of the 7th Computer Assisted Assessment Conference, 2003.
- Mohler, M., Mihalcea, R., 2009. Text-to-text semantic similarity for automatic short answer grading, In Proceedings of the European Association for Computational Linguistics (EACL 2009), Athens, Greece.
- Moodle, 2015. https://docs.moodle.org/20/en/Regular_Expression_Short-Answer_question_type, March
- Palmer, K., Richardson, P., 2003. On-line assessment and free-response input - a pedagogic and technical model for squaring the circle. In Proceedings of the 7th Computer Assisted Assessment Conference.
- Perez, D., 2007. Adaptive Computer Assisted Assessment of free-text students' answers: an approach to automatically generate students' conceptual models, phd thesis, Computer Science Department, Universidad Autonoma de Madrid.
- Ziai, R., Ott, N., Meurers, D., 2012. Short Answer Assessment : Establishing Links Between Research Strands. In *The 7th Workshop on the Innovative Use of NLP for Building Educational Applications* (pp. 190–200). Association for Computational Linguistics

