An Automatic Grading System Based on Dynamic Corpora

Djamal Bennouar
Department of Computer Science, Bouira University, Algeria

Abstract: *Assessment is a key component of the teaching and learning process. 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 order to enhance the learning process quality and the global student evaluation process and to highly reduce the assessment time and difficulties, most Algerian Universities were provided with an e-learning environment as a result of a government initiative. Unfortunately, such environment seems to be rarely used in the student's assessment process mainly due to the inefficiency of its Automatic Grading Subsystem (AGS) and the underlying corpora. A corpora used in the grading process contains a great number of miscellaneous answers, each one graded by more than two experts. Building efficient corpora for a course is actually a challenge. The underlying subjectivity in grading answers may have a serious impact in the corpus quality . The 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 a short answer AGS which has the capacity to dynamically build an up to date corpus related to each correct reference short answer. The automatically generated corpus is mainly based on a variety of indications specified by the teacher for each reference short answer. The early experiment of the presented AGS has shown its high efficiency for the automatic answers grading in some computer science courses.*

Keywords: *Architectures for educational technology system, country-specific developments, distance education and e-learning, evaluation methodologies, Computer Aided Assessment (CAA), AGS, Short answer, corpus, Answers predicting, text similarity.*

Received February 27, 2017; accepted May 10, 2017

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 [6, 20]. Two main kinds of assessment techniques are commonly used: the assessment based on selection-type question (also called objective-type questions) and the 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 by composing the answer in his/her own words and style. Even though the writing and the textual form are the most common, there are other ways to present data and ideas for open-ended questions like diagrams and 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 [17, 44]. 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 [44]. Essay questions, in contrast, are not limited in length, and students have the freedom of responding 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 [37]. 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 science learning processes 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 answers, a student has to stir his brain in order to show his capabilities and skills. As an example, in computer architecture, programming languages, 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 function's objectives, discover the function's logic, decide 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 disciplines use open ended questions to assess their students. In technological disciplines such as computer science, electronics, and civil engineering; short-answer questions seem to be the most used, while in social science such as juridical science and psychology; essay-type questions represent the main approach. The reason given by teachers using open ended questions is that selection type questions are not adapted to assess the 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 on the topic.

Writing answer to an open ended question actually requires more thinking than in answering a selection-type question, since the students must construct their own coherent answers and justifications [37].

Actually, the two arguments cope with the result of many research activities. Sigel [41] 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 researches [7, 24, 27, 34] open ended questions represent the efficient approach if the teacher target to assess the student ability to synthesize and analyze the 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 students 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 students. Written text is usually full of misspelling and wrong sentences construction. Homophone represents one of the main errors sources in text writing.
- The great students number who ask for the consultation phase. In Algerian Universities, the assessment of a student answers goes through two phases: In first phase the teacher evaluates the student's answers and grades them. In a second phase students are invited to consult their answers and compare them to the correct ones provided by the teacher. This second phase is time consuming and difficult to achieve. Often, for each student, the teacher has to undertake a hard dialog. Usually, the teacher reread with the student the answers, takes into account the student's arguments, does his best to convince the student that some part in the answer are undeniably false 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 the assessment time and difficulties, most Algerian Universities were provided with an e-learning environment as a result of a (or according to) government initiative. An e-Learning environment is usually represented by at least one classroom provided with a number of desktop computers connected to a server machine hosting a Learning Management System (LMS). Unfortunately, such environments seem to be rarely used in the student's assessment process. As an example, Bennouar [5] reported that in the Blida University, which counts 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 students using selection-type questions. From a scientific and technical point of view this situation is due to the main following reasons according to [5]:

- Like some skeptical researchers [37], many teachers are not convinced that Automatic Grading Subsystem (AGS) have the capacity to efficiently grade free text answers for their courses, even for short answers.
- Automatic grading tools for short answers, included in the provided LMS platforms, are either not efficient to support assessment [5] in an hostile environment or are not simple to use even by computer science teachers.

According to [5], providing an efficient AGS for short answers represents a key element to make the teachers interested by the e-Learning environments in their learning processes. An efficient AGS for short answers has to be provided with:

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

This paper presents an AGS which is planned to be progressively adapted for various environments even for hostile ones. The key concepts of such tools are its capacity to be tuned by assessors to deal with: the various courses kinds, the short answer questions' objectives, the student's knowledge and skills and the student's language level. An efficient AGS means that: on the one side the student's answers receive the correct grade and on the other side the number of students claiming for a manual review of their answers has to be very limited. These objectives are mainly reached by the capacity of the AGS to dynamically build for each reference short answer, an up to date corpus based on the teacher's indications.

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 questions kind (enumeration, choice justification, definition, description etc.,), the techniques used to specify and build the model answer, the chosen approach to achieve the comparison process, the resource needed for the comparison process, and the kind of expected assessment results differ from one approach to another. In addition, many approaches seem to be heavily impacted by a specific language and domain [37].

The comparison process which represents the core of such approaches is highly related to the task of text similarity [28] which is essentially the problem of detecting and comparing the features of two texts. The various techniques used in this process may be classified in two main categories: techniques based on pure string similarity and techniques based on semantic similarity. String similarity approaches are characterized by their simplicity and language-independence while semantic similarities are usually language and domain dependent. Such two approaches may also be combined in order to get sometimes more interesting results [36].

2.1. String Similarity

Two string similarity kinds are used in short answers grading: full string similarity and similarity distance. Full string similarity represents the simplest technique used to grade a student answer. In this naïve technique the student answer has to exactly match one of the Reference Answers (RA) in order to get its associated score. If no match is detected, the answer is 0 graded. This kind of similarity is achieved at the character level.

The similarity distance is a number representing an algorithm-specific indication of similarity or dissimilarity between a student answer and a reference answer. Grading the student answer depends on this distance. This technique is intensively used in many areas like information integration, information retrieval, document clustering and so on. In this technique, a measure may be achieved either at character level (character-based similarity measures) or at term level (term-based similarity measures).

2.1.1. Full Similarity Techniques

Despite their simplicity, full similarity techniques are actually used by short answer tools deployed in the context of well-known open source LMS platforms such as Moodle, eFront and Claroline. Such tools perform a simple string comparison between the student's answer and a set of answers defined by the teacher. Each teacher's answer, specified as a single line, is associated with a grade spanning from 100% to 0%. In Moodle, the teacher may specify if the comparison has to be case sensitive or not. Such tools have to be used in situations where an answer has to fully match one of the provided teacher's answers in order to get the associated grade , otherwise the grade is 0. Actually, there are many domains where such tools may be efficient. In domains, like religious science, poetry, computer programming, and communication protocol languages even a small error is not accepted. Despite the exact nature of the expected answer, teachers face actually many difficulties with these tools. As an example, in Moodle short-answer tool, an extra blank character may lead to an error. The teacher has to provide students with many advices in order to let them write the answers very carefully.

Tools enabling more flexibility make the use of regular expressions in the specification of model answers. Actually, the main goal of regular expressions is to compress in a single expression a great number of alternatives and equivalent answers. Moodle Regex [30] and WebLas [3] are kinds of such tools. In Moodle Regex, the teacher has to specify manually the answer as a regular expression. All regular expressions power may be used in specifying the model answers. The main drawback of this tools kind is the difficulty to understanding the regular expression concepts and the huge difficulty to operate them. Even computer science teachers find difficulties with regular expression specifications.

In WebLas [3], a limited regular expression is created automatically for each model answer after determining, under the control of the teacher, 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 the regular notation himself. According to [3], WebLas regular expressions are very simple since they are built using only the alternative operator and seem to be oriented to deal with a specific assessment kind (English language ability). In addition, according to [44] the WebLas misses an evaluation study based on data.

2.1.2. String Similarity Metrics

Looking for string similarity and defining metrics for such similarity were first introduced in domains other than Computer Assisted Assessment (CAA), like information retrieval, indexing, translation systems and so on. The Vector space model ([39] seems to be one of the earliest approaches dealing with text similarity for automatic indexing. The similarity metric used by Support Vector Machine (SVM) is the cosine coefficient which measures the angle between a vector representing a reference document and another vector representing a query.

Nowadays, a great number of string similarity algorithms and metrics exist and some of them are now used in short answer grading. Usually to evaluate the efficiency of a short answer grading approach, Pearson's correlation coefficient measure is used in order to show the grading tendency of an approach regarding an average of human grades. In addition to Pearson correlation, Noorbehbahani [32] uses adjacent agreement calculation to measure the number of times that the system score and the human score differ up to one point.

The BLEU algorithm [33], originally designed to evaluate automatic Machine Translation systems is used in [35] to assess short student’s essays. Perez [35] has shown that for a specific kind of questions category the BLEU algorithm attains better results than other keyword-based procedures. The BLEU algorithm calculates an n-gram metric and applies a brevity penalty factor to penalize the short texts. According to [35] the modification, the Brevity Penalty Factor may lead to more interesting results especially when unigram metric is used.

Many decorations were introduced to enhance the BLEU algorithm performance [32, 43]. Actually, these decorations put the modified Bleu algorithm in the border line between string similarity and semantic similarity. Globally, the decoration takes into account text characteristics.

The M-BLEU which is argued to be a modification of the BLEU algorithm [32] associates weights to words to show their importance, searches for synonymy between non matched words in reference answer and student answer then modifies student answer by the replacement of a word with its synonym if this later exists in the reference answer, and take into account the word importance (n-gram weight) in the brevity factor calculation. Noorbehbahani [32] claims that the evaluation of his M-BLEU process showed 0.85 correlations and an adjacent agreements of at least 75%. A recent modification of the BLEU algorithm in a machine translation research work (Wolf 2014) deals with two text characteristics: synonymy and rare word.

Automatic Short-Answer Grading System (ASAGS) system described in [40] used a number of text characteristics such as synonymy, numeric value match (e.g.,10th and tenth), adjectival and demonymic forms for countries and nations, and acronym match. For three kinds of questions (definitions, advantages or disadvantages and Yes or No questions with justification), Selvi [40] claimed that the ASAGS system obtained a high correlation value.

In [14] thirteen popular similarity algorithms were tested on short answers using the texas short answer data set [29]. The best correlation value (0.435) was obtained by mixing bi-gram and tri-gram similarity measures applied to a raw text. The preparation of texts by stemming and removing stop words, did not introduced appreciable changes in the results.

The system proposed by [31] may also be positioned in the borderline of semantic similarity and string similarity. Starting with a model answer, the system of [31] uses a synonym dictionary and generates a number of equivalent model answers before using three string similarity algorithms: the Common Words (COW), Longest Common Subsequence (LCS) and a specific similarity called semantic similarity which is actually a specific string similarity applicable to text sentences. The proposed system was compared to ASAGS [40] and the authors claimed that their system outperformed ASAGS and the well-known Latent Semantic Analysis (LSA) algorithm [23].

2.2. Semantic Similarity

Semantic similarity is usually achieved based either on a corpus or a knowledge base [15]. corpus-based similarity determines the similarity between words according to information gained from a large collection of texts that is used for language research. knowledge-based similarity measures the similarity degree between words using information derived from semantic networks. WordNet represents the most popular semantic network where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations [15].

The techniques used for semantic similarities approaches were classified by [26] in three main categories: statistical, information extraction and full natural-language processing. Perez [37] added two categories: clustering and semantics network comparison. Such techniques were originally introduced to deal with essay and are not necessarily well suited for short answer grading.

Paraphrase recognition approach used in C-Rater (Leacock and Chodorow, 2003) was tested on answers having an average of 43 words, which may be considered as a small essay and on answers having an average of 15 words. In both cases, the agreement percentage between C-rater and the human experts reached 84%.

Information Extraction (IE) techniques backed with some Natural Language Processing (NLP) techniques dealing mainly with misspelling and phrase chunking into nouns and verbs, were used in academic tools such as Automark [26] and the Oxford's system reported in [38]. The reported agreement between experts and machine may reach 96% for AutoMark and 84% for the Oxford's system. The Automark's excellent agreement is actually due to the fact that four kinds of expected student's answers were used is the evaluation process: Single word, Single value, a short explanatory sentence and description of a pattern in data. It is clear that linguistic complexity for the latter is higher than the third which in turn is more complex than the two firsts. Agreement for single word and single value reached 100% while the maximum agreement obtained for the fourth kind of answers was 87%.

Many studies have shown that, in opposite to NLP and IE techniques, statistical techniques represented mainly by the well-known LSA [23] and the Probabilistic LSA [19] are not well suited for the assessment of short answers [10, 22]. LSA underpins a number of academic and commercial tools like Apex Assessor [10], Research Methods Tutor (RMT) [42] and Intelligent Essay Assessor (IEA) [13]. Actually, a number of extended reviews on semantic similarity approaches, as well as their related techniques and tools may be found in [15, 37, 44].

2.3. Combination of Similarities

Almost all short answer assessment systems deal with text written in English. Only a few works target other language such as German, Japanese and French [37, 44]. The language dependencies represent actually one of the major drawbacks of current systems based only on semantic similarity approaches. Adapting such systems to another language requires the introduction of significant modifications. Replacing language-dependent processing modules was the strategy adopted by [25] to realize the CoMiC-DE German version of CoMiC-EN. Providing short answer assessment systems with a translation tool from a targeted language to English is another strategy adopted by [1, 16]. With this later strategy, the authors reported that the loss of accuracy is small.

To insure language independence and to enhance the similarity measure, many researches adopted the idea of combining string and semantic similarities techniques [4, 9, 16, 18, 31, 36].

Perez *et al*. [35] has combined a modified BLEU algorithm called ERB and the LSA. The experiments have shown that the combinations always perform better than their constituent techniques and slightly better results may be obtained by assigning an optimized weight to each technique.

To show the importance of including string similarity in the process of calculating short answer similarity, Islam and Inkpen [22] reported situations where good similarities are obtained only if string similarity is used. Neither dictionary-based similarity nor corpus based similarity can obtain interesting results. The overall similarity calculated by [22] combines three modified version of the LCS [2] and a Corpus Based similarity called Second Order Co-occurrence Pointwise Mutual Information (SOC-PMI) described in [21]. According to [22], the proposed combined approach called e Semantic Text Similarity (STS) achieves a very good Pearson correlation and outperforms the results obtained by the previous work which use larger corpus and semantic networks. In addition, lower time complexity of STS seems to be one of its main advantages compared to other previous works. STS uses only one corpus-based measure, while others combine both corpus-based and WordNet-based measures.

Heilman and Madnani [18] reported results which show that a combination of a great number of approaches do not always produce notable enhancement. Heilman and Madnani [18] tested a first system combining n-gram and a semantic approach and a second system combining the first system, the BLEU algorithm and a semantic similarity of short texts pair called PERP [18]. The obtained results show that the second system did not always outperform the first one. A similar remark was also revealed by the works of [9].

Actually, the strategy of combining string similarity with semantic similarity is also adopted by a number of researches focusing on text semantic similarity. The system proposed by [9] combines the n-gram based similarity measure called Clustered Keywords Positional Distance (CKPD) [8] and a concept similarity using WordNet called "ProxiGenea" [9].

The tests conducted have shown that the results depend heavily on the test sets used. The combination produced best results with only some test sets. With other test sets, the best results were obtained either by the semantic similarity alone or the n-gram based similarity alone.

Gomaa and Fahmy [16] combined n-gram string similarity and the DIStributionally similar words using CO-occurrences (DISCO) corpus based similarity [11]. In order to show the effectiveness of the proposed combination, Gomaa and Fahmy [16] conducted his experiments in three steps. In a first step, 13 String-Based similarity algorithms were evaluated. The best correlation values of 0.435 were obtained with the n-gram algorithm. In a second step two similarity measures of DISCO, called DISCO1 and DISCO2 were used. The best correlation value of 0.465 was obtained using DISCO1. In the third step, a number of combination between string based similarity algorithm and the two DISCO similarities were tested. The best correlation value (0.504) was obtained from mixing N-gram with DISCO1.

3. Challenges of Short Answer Automatic Grading

Today, it appears clearly that most approaches dealing with the text similarity problem has reached a high yielding tools maturity degree which may be highly efficient in many field, such as machine translation, natural language understanding, sentiment analysis and so on. In this kind of fields, the results may be efficiently used even if they are not the exact awaited ones or are judged as medium. Such kinds of incomplete results may be considered as a good starting point either to take a decision or to initiate an easy and rapid adjustment and enhancement of the results' process.

In the field of grading answers to open ended questions, it seems that this maturity has not yet reached a level where a grading approach may be used efficiently in a university context, more precisely, in an Algerian University; due to the environment hostility and also to some important issues not covered by current approaches and tools.

3.1. The Environment Hostility

Unlike some other related fields, the results of a text similarity process in grading answers to open ended questions are decisions and judgments which have to be either accepted or rejected by the students. The rejection is not a starting point for other constructive activities but for a manual assessment task where teacher and students have to provide additional information, arguments and explanation in order to adjust or confirm the grade. The number of rejects in a university context has to be minimal in order to preserve the efficiency of an automatic grading approach. Current AG approaches and tools are usually evaluated in a favorable environment characterized by two main aspects:

- The presence of corpora and knowledge bases.
- The answers provided by teachers and students are usually correctly written and in some rare cases, 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 answers grading. As an example, the spelling errors and homophones based errors represent a common situation in student's answer and even some answers provided by teachers may also contain errors.

The lack for corpora or knowledge base ready to be operated by any grading approach or tools is another characteristic of the hostile environment. To our 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 [28]. In the dataset used by [28], some grades reported by experts differed in some situations 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 a student who is 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 the 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 errors. This tolerance will be reduced in the 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.

This environment hostility is not the sole challenge AG approaches and tools are facing. Actually, current approaches and tools suffer from a number of lacks which make very hard their direct reuse in Algerian Universities and most third world universities.

3.1.1. Courses Specificity

In a University context, an e-learning system is usually provided as an aid for the learning process of any course and is intended to be a platform to be used for assessing students. Current assessing approaches or tools are specific for a course kind and even for a question kind of. Even, if we integrate all existing tools in the e-learning platform, this later will not cover a small range of the university courses.

3.1.2. Language Dependence

More than one language may be used in the teaching activities. This is the case for Algerian Universities, where two main languages are used: Arabic and

French. Existing approaches and tools seem to be very specific for a kind of language. Tools combining string similarity and semantic text similarity are actually strongly coupled to language specificities, since the combination still use corpora or knowledge bases related to a specific language. Even, the translation phase used by some combined grading approaches to overcome the language dependency can't actually enhance the final results. As reported in [15], the translation part, which doesn't always guaranty a good result, has a negative impact in the semantic part of the grading system.

3.2. Issues Not Covered By Current Approaches

- *Grade rejects minimization*: As reported before, the result of an AG process is a decision which may be not accepted by students. Minimizing the reject of an automatic grading process seems to be a not considered topic by current approaches. The high correlation metric obtained by some approaches in some specific conditions cannot directly inform about the reject amount in a grading process.
- *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 or a subject name when composing a small sentence, etc., Such words and values become accessible when the student submits his answer to the AGS. 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 by a student during the answer elaboration process. 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 differentiates between dynamic and static contents. To our knowledge none of the current corpora, approaches and tools deals with such answers kind.
- *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 parts 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 answers. Multiple short answers technique has to take into account many challenges such as the impact of additional non awaited short answers and multiple report of a same short answer.
- *Teacher preferences and orientation*: None of the proposed approaches for automatic assessment take 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 or not, 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 parts. The major part of the recent research focus on enhancing the similarity process without taking into account the fact that in addition to the model answer, a teacher can provide advices, hints, constrains and other information which have to be taken into account in the process of evaluating an answer.

4. The proposed AGS

4.1. System Overview

Figure 1 reports the AGS global architecture which clearly shows that the grading process goes through two important phases: the preparation phase and the grading phase. The preparation phase achieves the most important and intelligent works in the system in order to make efficient the grading phase which is mainly based on string similarity techniques.

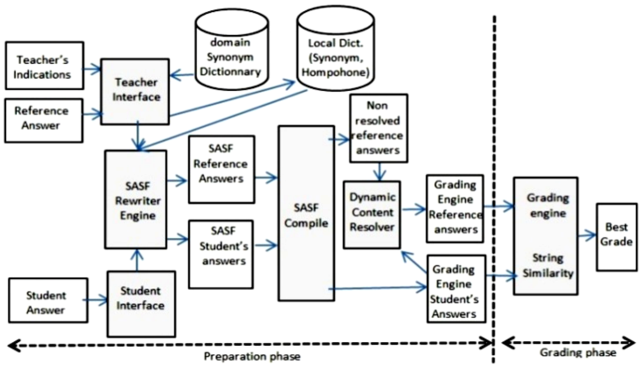

Figure 1. AGS architecture.

In the process of preparing a question, a teacher submits to the preparation phase a set of 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 grades (Source code 1). 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.

- *Source code 1*: Corpus Generation Core Module

```
// Corpus generation: used to generate the GERA and GESA
corpus
// for a single reference answer
// Inputs
//   - ra: represents a reference answer
//   - ig: initial grade associate with ra
//   - tios: an ordred set of teacher indications
public Corpus buildCorpus(String ra, float ig,
                                TeacherIndicationOrdredSet tios){
  // Step 1: Create a first empty corpus, called sc ( source
corpus)
  Corpus  sc = new Corpus();
  // Create a second empty corpus, called tc (target corpus)
  Corpus  tc = new Corpus();
  // initialize the source corpus with the reference answer
  // and associated initial grade
  CorpusElement ce = buildCorpusElement(ra, ig)
  sourceCorpus.addCorpusElement(ce);
  // Copy the content of sc to tc
  tc.addAll(sc)
  //Apply successively the teacher indications to each source
corpus
  // element. Each indication application yield new corpus
elements
  // which are added to the target corpus tc
  for (TeacherIndication  ti: tios ){
     for (CorpusElement ce: sc)
        tc.addCorpusElement(applyIndication(ti, ce)
        // next indication must be applied to all existing corpus
        // elements which are in tc. for next iteration initialize sc
with
        // the content of tc
        sc.initialize(tc)
   }
  // return the final corpus which is now completely built
   return tc;
}
```

The grading process tries 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).

4.2. Specification of RA and Teacher Indications

Once the RA 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 RA tagged with the indication penalty factor. Through the indications, the teacher tries to predict the various student answers forms 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 RA specification and their associated indications is achieved using an XML based language called the Short Answer Specification Language (SASL) [12]. The teacher indications are represented by specific XML tags and regular expression meta-characters. The main objective of the specification of RA accompanied with a number of indications is to efficiently guide the preparation and the grading phases.

The use of XML tags and regular expressions, even by the computer science teachers is not easy at all. Like other XML based systems, 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 SASL 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 (Table 2).

Table 1. Sample SASL common annotations.

SASL 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 2. A reference answer specification with dynamic content and keywords.

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 first name 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:{

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 3 reports the main indications classes supported in the current version of SASL and Table 4 shows the definition of more indications to be applied for the answer shown in Table 2.

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 teacher control. 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.

Table 3. A sample of teacher Interface sub controllers and SASL indications classes.

Indication Classes Controller	Role
Grading controller	Define maximum grade and similarity thresholds, select similarity tools an 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 4. Example of indications for the question in Table 2 specified using teacher Interface.

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%

4.3. The SASL Compiler

The student answers, the RA 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 [12]. An SASL 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 the student answers.

The specified indications are not applied systematically to the RA 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 pre-process 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.

The main task of the SASL compiler is the production of an ordered set of reference answer called the GERA (Table 5) and an ordered set of student answers called the GESA (Table 6). 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 pre-process 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.

Table 5. A Sample of compiled non resolved RA.

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 Persone(String $alpha-id1,String $alpha-id2){	5
6	Personne(String $alpha-id1,String $alpha-id2){	10
11	pub Persone(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

When a reference answer contains a dynamic content, the SASL compiler produces a set of non-resolved RA (Table 5) which needs to be submitted to a resolution process in order to produce the GERA. The dynamic content resolution process, called the answer dynamic content resolver, is activated when a student answer is compiled. The Answer Dynamic Content Resolver tries 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 Tables 5 and 6, 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.

4.4. The Grading Phase

The grading engine may be provided with a number of text similarity tools which may be successively applied according to the teacher indications. The process followed by each text similarity tools is the same. The process considers one by one the answers of the GESA ordered set, according to their position in the set. For each considered student answer (Table 6), the tools try to locate the most similar answers in the GERA ordered set. When the comparison produces a result greater than the Manual Grading Threshold (MGT), the answer is tagged in order to be submitted to manual review. If the comparison yields a value greater or equals the Automatic Grading Threshold (AGT), the answer is graded with the grade of the GERA answer and any eventual tag recommending a manual review is then discarded.

Table 6. Some answers from the GERAs.

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 7.Result with Automatic Grading Threshold equals to 97%.

GESA ID	GERA ID	N-Gram Sim 97%	Grade penalty
4	26	0.9736842	50
6	32	0.9705882	55
7	34	0.972973	55
10	19	0.974359	45
10	26	1.0	50
12	25	0.9714286	50
12	32	1.0	55
13	29	0.9736842	50
13	34	1.0	55

Once terminated, the grading engine comparison process keeps only the answer satisfying the automatic grading threshold (or manual grading threshold if no comparison reached the prefixed AGT). Among those answers, the grading engine chooses the GERA answer having the smallest position in the GERA answer set and assign its grade to the student answer. According to the GERA organization, this answer is the one having the smallest penalty among the kept answers.

Table 8. Results with automatic grading threshold equals to 98%.

GESA ID	GERA ID	N-Gram Sim 98%	Grade penalty
10	26	1.0	50
12	32	1.0	55
13	34	1.0	55

As an example, Tables 7 and 8 report the result of a comparison between the GESA in Table 6 and the GERA in Table 5, using an n-gram based algorithm. The Automatic Grading Threshold was positioned to 97% for Table 7 and 98% forTable 8. The retained answer is the one having the smallest position in the GERA (19 in Table 7 and 26 inTable 8).

5. Experimental Results

The AG system described in this paper is currently, under development as a plugin in the context of Moodle LMS. Testing the system in a real environment is actually a challenge in Algerian Universities. Despite the fact that many universities provide at least one LMS aimed to be used in learning processes, automatic assessment tools found in such LMS seems to be not exploited at all by teachers. In order to test the efficiency of our system, we have technically supported some teachers from the University of Bouira (Algeria) to use Moodle Short Answer automatic grading plugin to assess student in some computer science courses.

Moodle short answer plugin provides for users the facility to manually specify a corpus for a question. The corpus is a set of correct or partially correct RA. The comparison process of the Moodle short answer is based on an exact match between a student answer and one of the specified references answers. Hence, if the student answer and the reference answer differ only by one character (a space or an accent for example), the comparison process fails and the plugin consider the student answer as a non-correct answer.

To try to efficiently operate the moodle short answer plugin, a number of 32 recommendations were written for students. Those later have to deeply understand and rigorously apply the recommendation when writing their answers. Some recommendations focus on the answer structure while others focus on the character arrangement in an answer. The recommendations related to the answers structure actually depends on the course targeted by the automatic grading tool. As an example, in the course titled “Object Oriented Programming using Java”, each java language instruction should be written according to a specific java programming style. The recommendation related to the answer character arrangement is mainly due to the comparison process of the moodle short answer plugin. As an example, some recommendation specifies the exact number of character space to use in order to separate the words, or to separate a special character from an alphabetic character, or to separate a special character from another special character. In the same kind of recommendations, we can find a recommendation which prohibits the use of accent in an answer.

To test the efficiency of our system, we used many answers sets graded by moodle short answer. the set of answers and their evaluation were constructed during the use of moodle short answer plugin to evaluate the students' answers in three computer science courses: object oriented programming using java, algorithm and data structure and computer networks. the student evaluation using moodle short answer was conducted in the context of the bouira university examination moodle platform site[1].

For this early system test, only three indications kinds were used to generate the system corpora (a corpus for each reference answer): ignoring extra space, ignoring accent and accept ordered partial keywords. The comparison process activated is similar to the one used in moodle short answer (the AGT were set to 100% to reach exact match when comparing two texts). The results reported in Table 9 show that a great number of answers were not correctly evaluated by moodle short answer even with the use of written recommendations for answer. two main reasons were behind moodle short answer incorrect evaluation: the difficulty for a great students number to correctly apply the recommendations and to correctly write the keywords or parts of them. Actually, since the recommendations number was important, a non-neglected pressure was put on a student when he elaborates an answer. The column labeled *tests* represents the electronic exam number evaluated using moodle short answers. the column titled Adjusted tests reports the number of exam salvaged by our system. Salvaged exam were not correctly or accurately evaluated by moodle short answer. The Lines 3Bis and 4Bis report the second round results of the exam reported in lines 3 and 4. The second round concerns students who failed in the first round. The contents of the first and the second round exam were exactly the same. We notice that the number of salvaged attempts is important with students who failed in the first round.

Table 9. Comparison between proposed AGS and moodle shortanswer associated with written recommendations.

Id	Course ID	Date	tests	Adjusted tests	adjusted tests rate
1	OOP-J	02/2017	104	38	37%
2	CN	01/2016	73	12	16%
3	OOP-J	01/2016	44	13	30%
4	ADS	01/2016	96	58	60%
3Bis	OOP-J	02/2016	41	17	41%
4Bis	ADS	02/2016	20	14	70%

6. Conclusions 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 RA to produce a rich corpus containing answers with their associated grade.

This approach is currently, under implementation in the context of the moodle LMS platform. The first experiments of some system parts (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 students in 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 question kind and robust regarding the various kinds of frequent language error, mainly accent, missing characters and homophone.

Future work will focus on completing the miscellaneous aspect of the SASL 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 answers 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

[1] Alfonseca E. and Pérez D., "Automatic Assessment of Open Ended Questions with a Bleu-Inspired Algorithm and Shallow NLP," *Advances in Natural Language Processing*, vol. 3230, pp. 25-35, 2004.

[2] Allison, L. and Dix T., "A Bit-String Longest-Common-Subsequence Algorithm," *Information Processing Letters*, vol. 23, no. 5, pp. 305-310, 1986.

[3] Bachman L., Carr N., Kamei G., Kim M., Pan M., Salvador C., and Sawaki Y., "A Reliable Approach to Automatic Assessment of Short Answer Free Responses," *in Proceeding of the 19th International Conference on Computational Linguistics*, Taipei, pp. 1-4, 2002.

[4] Bar D., Biemann C., Gurevych I., and Zesch T., "UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures," *in Proceeding of the 19th*

[1] http://examen.univ-bouira.dz

international conference on Computational linguistics, Montreal, pp. 435-440, 2012.

[5] Bennouar D., "Challenges of e-Test in a University Context," *in Proceeding of 1st Elearning Spring School*, Algeria, pp. 27-30, 2013.

[6] Biggs J. and Tang C., *Teaching for Quality Learning at University*, McGraw-Hill, 2011.

[7] Birenbaum M., Tatsuoka K., and Gutvirtz Y., "Effects of Response Format on Diagnostic Assessment of Scholastic Achievement," *Applied Psychological Measurement*, vol. 16, no. 4, pp. 353-363, 1992.

[8] Buscaldi D., Rosso P., Gomez J., and Sanchis E., "Answering Questions with an n-Gram Based Passage Retrieval Engine," *Journal of Intelligent Information Systems*, vol. 34, no. 2, pp. 113-134, 2009.

[9] Buscaldi D., Tournier R., Aussenac-Gilles N., and Mothe J., "IRIT: Textual Similarity Combining Conceptual Similarity with an N-Gram Comparison Method," *in Proceeding of the 1st Joint Conference on Lexical and Computational Semantics*, Montreal, pp. 552-556, 2012.

[10] Dessus P., Lemaire B., and Vernier A., "Free Text Assessment in a Virtual Campus," *in Proceeding of the 3rd International Conference on Human System Learning*, pp. 61-75, 2000.

[11] Disco, http://www.linguatools.de/disco, Last Visited 2017

[12] Djelattou R., "A Short Answer Specification Language for an Automatic Grading System," M.S. Thesis, the Saad Dahlab University, 2015.

[13] Foltz P., Laham D., and Landauer T., "The Intelligent Essay Assessor: Applications to Educational Technology," *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, vol. 1, no. 2, pp. 1-5, 1999.

[14] Gomaa W. and Fahmy A., "Short Answer Grading Using String Similarity And Corpus-Based Similarity," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 11, pp. 115-121, 2012.

[15] Gomaa W. and Fahmy A., "A Survey of Text Similarity Approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13-18, 2013.

[16] Gomaa W. and Fahmy A., "Automatic Scoring for Answers to Arabic Test Questions," *Computer Speech and Language*, vol. 28, no. 4, pp. 833-857, 2014.

[17] Heinrich E., "Addressing Efficiency and Quality of Marking in Essay Assessment with E-learning Support," *Journal of Distance Learning*, vol. 10, no. 1, pp. 15-25, 2006.

[18] Heilman M. and Madnani N., "HENRY-CORE: Domain Adaptation and Stacking for Text Similarity," *in Proceeding of the 2nd Joint Conference on Lexical and Computational Semantics*, Atlanta, pp. 96-102, 2013.

[19] Hofmann T., "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine Learning*, vol. 42, no. 1, pp. 177-96, 2001.

[20] Iahad N., Dafoulas G., Kalaitzakis E., and Macaulay L., "Evaluation of Online Assessment : The Role of Feedback in Learner-Centered e-Learning Learner-Centred Paradigm," *in Proceeding of the 37th Hawaii International Conference on System Sciences*, Big Island, pp. 1-10, 2004.

[21] Islam A. and Inkpen D., "Second Order Co-Occurrence PMI for Determining the Semantic Similarity of Words," *in Proceeding of the International Conference on Language Resources and Evaluation*, Genoa, pp. 1033-1038, 2006.

[22] Islam A. and Inkpen D., "Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 2, pp. 1-25, 2008.

[23] Landauer T. and Dumais S., "A Solution to Platos Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge," *Psychological Review*, vol. 104, no. 2, pp. 211-40, 1997.

[24] Mcgrath P., "Assessing Students: Computer Simulation vs MCQs," *in Proceeding of the 7th Computer Assisted Assessment Conference*, Loughborough, pp. 243-246, 2003.

[25] Meurers D., Ziai R., Ott N., and Kopp J., "Evaluating Answers to Reading Comprehension Questions in Context: Results for German and the Role of Information Structure," *in Proceeding of the TextInfer Workshop on Textual Entailment*, Scotland, pp. 1-9, 2011.

[26] Mitchell T., Russell T., Broomhead P., and Aldridge N., "Towards Robust Computerised Marking of Free-Text Responses," *in Proceeding of the 6th Computer Assisted Assessment Conference*, Loughborough, pp. 233-249, 2002.

[27] Mitchell T., Aldridge N., Williamson W., and Broomhead P., "Computer Based Testing of Medial Knowledge," *in Proceeding of the 7th Computer Assisted Assessment Conference*, Loughborough, pp. 249-267, 2003.

[28] Mohler M. and Mihalcea R., "Text-to-Text Semantic Similarity for Automatic Short Answer Grading," *in Proceeding the 12th Conference of the European Association for Computational Linguistics*, Athens, pp. 567-575, 2009.

[29] Mohler M., Bunescu R., and Mihalcea R., "Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency Graph Alignments," *in Proceeding of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, pp. 752-762, 2011.

[30] Moodle, https://docs.moodle.org/20/en/Regular_Expression_Short-Answer_question_type, Last Visited 2017.

[31] Muftah A. and Juzaiddin M., "Automatic Essay Grading System For Short Answers In English Language," *Journal of Computer Science*, vol. 9, no. 10, pp. 1369-1382, 2013.

[32] Noorbehbahani F. and Kardan A., "The Automatic Assessment of Free Text Answers using a Modified BLEU Algorithm," *Computers and Education*, vol. 56, no. 2, pp. 337-345, 2011.

[33] Papineni K., Roukos S., Ward T., and Zhu W., "Bleu: a Method for Automatic Evaluation of Machine Translation," *in Proceeding of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, pp. 311-318, 2001.

[34] Palmer K. and Richardson P., "On-line Assessment and Free-Response Input- a Pedagogic and Technical Model for Squaring the Circle," *in Proceeding of the 7th Computer Assisted Assessment Conference*, pp. 289-300, 2003.

[35] Perez D., Alfonseca E., and Rodríguez P., "Application of the Bleu Method for Evaluating Free-Text Answers in an E-Learning Environment," *in Proceeding of the Language Resources and Evaluation Conference*, Lisbon, pp. 1351- 1354, 2004.

[36] Perez D., Gliozzo A., Strapparava C., Alfonseca E., Rodriguez P., and Magnini B., "In Automatic Assessment of Students' Free-Text Answers Underpinned by the Combination of a BLEU-Inspired Algorithm and Latent Semantic Analysis," *in Proceeding of the 18th International Florida Artificial Intelligence Research Society Conference*, Florida, pp. 358-362, 2005.

[37] Perez D., "Adaptive Computer Assisted Assessment of Free-Text Students' Answers: An Approach to Automatically Generate Students' Conceptual Models," Ph.D. Thesis, Universidad Autonoma de Madrid, 2007.

[38] Pulman S. and Sukkarieh J., "Automatic Short Answer Marking," *in Proceeding of the 2nd Workshop on Building Educational Applications Using NLP*, Ann Arbor, pp. 9-16, 2005.

[39] Salton G., Wong A., and Yang C., "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613-620, 1975.

[40] Selvi P. and Bnerjee A., "Automatic Short-Answer Grading System (ASAGS)," *International Journal of Computer Science and Networking*, vol. 2, no. 1, pp. 19-23, 2010.

[41] Sigel I., *Development of Mental Representations: Theories and Applications*, Psychology Press, 2013.

[42] Wiemer-Hastings P., Allbritton D., and Arnott E., "RMT: A Dialog-Based Research Methods Tutor with or without a Head," *in Proceeding of the Intelligent Tutoring Systems 7th International Conference*, Berlin, pp. 614-623, 2004.

[43] Wołk K. and Marasek K., "Enhanced Bilingual Evaluation Understudy," *Lecture Notes on Information Theory*, vol. 2, no. 2, pp. 191-197, 2014.

[44] Ziai R., Ott N., and Meurers D., "Short Answer Assessment : Establishing Links Between Research Strands," *in Proceeding of the 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, Montreal, pp. 190-200, 2014.

Djamal Bennouaris a Professor at the University of Bouira, Algeria, an Associate Researcher in the National Center for the Development of Advanced Technologies (CDTA), Algiers, the Director of the LIMPAF laboratory (Software System and Sensor Networks for Agriculture and Forestry) and a member of the LRDSI Lab at the SaadDahlab University of Blida, Algeria. He obtained the Magister degree fromthe National Institute for Computer Science (INI), Algeria, in 1993 and the Phddegree from the EcoleSuperieured'Informatique (ESI), Algeria, in 2009. In the CDTA,D. Bennouar conducted various research related to VLSI CAD Frameworks (HDL,Inter tools communication, Engineering Databases), Computer Networking and Software Product Lines for E-Government. Currently, his mainresearch interests include Software Architecture, Software Product Linesfor Agriculture and Forestryand Automatic evaluation of essays and short answers. He is supervising a number of PHD students preparing their thesis in Software Architecture, Software Product Lines and Automatic Student Assessment.