Deanship of Graduated Studies
Al- Quds University

FLASA: Fuzzy Logic based on Automarking Short-text
Answers

Prepared by: Muhammad Mahmoud Al-Amayreh

Supervisor: Dr. Labib Arafeh

M.Sc Thesis

Jerusalem- Palestine

محرم 1427
FEB 2006

FLASA: Fuzzy Logic based on Automarking Short-text Answers

Prepared By
Muhammad Mahmood Al-Amayreh

B.Sc.: Computer Science from Al-Quds University/ Palestine

Supervisor: Dr. Labib Arafeh

A Thesis Submitted in Partial Fulfillment of Requirements for the Degree of Master of Computer Science / Computer Science Department / College of Science and Technology /Al-Quds University

محرم 1427
FEB 2006

Department of Computer Science
Deanship of Graduated Studies
⊙⊙**Al Quds University**⊙

FLASA: Fuzzy Logic based on Automarking Short-text Answers

Prepared By: Muhammad Mahmood Al-Amayreh
Registration No:

Supervisor: Dr. Labib Arafeh
Co-Supervisor: 20011468

Master thesis submitted and accepted, date: 15-02-2006
The names and signatures of the examining committee members as follows:

1- Dr. Labib Arafeh Head of Committee _____

2- Dr. <u>Rashid Jayousi</u> Internal Examiner _____

3- Dr. Mahmoud Al - Sahib External Examiner _____

Jerusalem- Palestine

محرم 1427

FEB 2006

Declaration:

 I certify that this thesis submitted for the degree of master in the result of my

own research except where otherwise acknowledged, and that this thesis (or any

part of the same) has not been submitted for a higher degree to any other

university or institution.

Signed

Muhammad Mahmood Al-Amayreh
15/02/2006

Acknowledgements

I would like to thank **Dr. Labib Arafeh** for his invaluable assistance and guidance, without which this work would never have come to fruition.

I would like to thank **Dr. Ibrahim Afaneh** for his help in reviewing this thesis, and **Mr. Wa'eel Qadomie** for giving me the time to complete my thesis.

Finally I thank my family and friends especially my parents and my wife for their patience, support and encouragement.

Abstract

In this work, a literature survey of automarking field has been reviewed as well as the techniques applied in these systems. In addition, the type of questions that are suitable to be automatically marked, the description of the currently most used systems of free texts answers' systems, both for academic and commercial environments have also been covered. A comparison of all of these systems in compliance with the currently available evaluation metrics have been introduced.

In cases with questions like essay or short text answers, hard problems like Syntactic, grammatical, Rhetorical structure analysis, Topical content analysis and Synonyms problem "Similarity or Related Words" come to stage. Such types of problems have attracted many interested researchers and system developers to introduce several designs to solve these problems. In fact, few of them have focused their studies on solving similar words problem.

In this thesis, we have concentrated our study on similar words' problem of short-text answer, and the various methods to solve this problem. A new approach for solving similar words problem in short-text answer questions based on a fuzzy logic algorithm is presented. We have emphasizeed our study on Fuzzy Logic techniques, and how to use each part of this method to solve similar words problem, in a way to explore the suitability of using it to solve such problems.

A Multi (3-5) Input Single Output, MISO, system have been tested, after which a general MISO model have been proposed and tested. Several approaches have been examined. Comparisons between the various results obtained have been conducted, 90% by the instructors evaluation.

A promising result shows the adequacy of using the fuzzy logic approach to such types of questions. It is recommended that further elaboration, testing, and comparison with the other currently available approached and systems are required.

ملخص

في هذا العمل ، قمنا بمراجعة تاريخية لمجال التصليح الالي للأسئلة ذي الاجابات القصيرة، والتقنيات التي أسندت لهذه الانظمة، ونوع الاسئلة المناسب التي يمكن أن تقييم، و وصف لغالبية الانظمة المستخدمة حاليا في تقييم الاسئلة قصيرة الاجابة أو الاسئلة المقالية سواءً المنتجة من قبل المؤسسات الاكاديمية أو المؤسسات التجارية، و مقارنة جميع هذه الانظمة في الالتزام بمقاييس التقييم المعترف عليها حاليا.

فيما يتعلق بالتصليح الالي للأجابات مثل إجابات المقالات أو النصوص القصيرة فقد برزت مشاكل صعبة مثل التحليل النحوي و البلاغي و التركيبي، و تحليل المحتوى الموضوعي و كذلك تحليل المرادفات اللغوية. ان مثل هذه المشاكل قد جذبت اهتمام العديد من الباحثين و مطوري الانظمة لطرح عدة تصاميم تعمل على حل هذه المشاكل، و في الحقيقة، فقد قام عدد قليل من هؤولاء بتركيز دراساتهم على مشكلة المرادفات .

في هذه الدراسة، قمنا بتركيز البحث على مسألة المرادفات في إجابات النصوص القصيرة و كذلك على الاساليب المتنوعة في حل هذه المشاكل، حيث تم تقديم اسلوب جديد في هذا الشأن ، و هو يعتمد على مبدأ خوارزمية المنطق الضبابي. و تركزت الدراسة على أجزاء المنطق الضبابي و كيفية استخدام كل جزء منها في مشكلة الكلمات المترادفة بطريقة تستكشف إمكانية استخدامه في حل مثل هذه المشاكل.

تم إختبار نظام ما يسمى بتعدد المدخلات (3-5) ومخرج أو نتيجة و احدة (المخرجات) (MISO)، و بعدها تم إقتراح وإختبار نموذج MISO عام. وكذلك تم فحص عدة أساليب و تم مقارنة نتائجها، و كانت النتيجة 90% مقارنة مع تقييم المدرسين.

تشير النتائج الواعدة الى ملائمة استخدام مفهوم المنطق الضبابي في حل مثل هذه الانواع من المسائل. كما و يوصى بالقيام باهتمامات، وإختبارات و مقارنات مع الاساليب و الانظمة المتوفرة حاليا.

TABLE OF CONTENTS

CHAPTER 6……………….…………………..…..………. **118**

CONCLUSIONS AND FUTURE WORK

APPENDIX A……………….………………………..…….….. 122

GLOSSARY

TECHNICAL DETAILS

QUESTIONNAIRES

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Introduction

The **FLASA** "**F**uzzy **L**ogic in **A**uto-marking **S**hort text **A**nswers" algorithm is one of methods that reduce the similar words problems in essay and short text answers, that may arise during the design of auto-marking system. When the questions are like essay or short text answers, hard problems like Syntactic, grammatical, Rhetorical structure analysis, Topical content analysis and Synonyms problem "Similarity or Related Words" comes to stage.

Such types of problems have attracted many interested researchers and system developers to introduce several designs which tackle these problems, but few of them have focused their studies on solving similar words.

FLASA algorithm is represented by a set of rules and the way of these rules are fired or executed. The main concept of the FLASA algorithm is it approach in solving similar words problems, by putting weight for each similar word in the answer.

The aim of the proposed auto-marking similar words in short answer is to generate auto-marking short-text answer once the exam is completed. Here we will focus our algorithm in one part of the short answers problems which called similar or related words. A typical auto-marking system usually consists of the accuracy property. So the main judgment in the comparison between the auto-marking similar words is the accurate of the results.

The most of the auto-marking short answer tools solving the similar words problems by the linear way c-rater, which mean that if the main words or their similarity was found in the answer; the answer is correct without taking in consideration that may be similar words are not in the same level of the main words in the key answer.

After a discussion with different instructors "PhDs holders" at Al-Quds University as shown in appendix C.1, about the way of evaluating the short answer questions, the following results has come out:

- Most of instructors do not pay attention to spelling problems, thus no marks go to right spelling neither dedicated marks for wrong spelling.

- The majority of instructors do not consider the Syntactic Structure Analysis and Rhetorical Structure Analysis problems of the phrases.

- Most instructors concentrate on finding the main words or related words within the student answer.

- Some instructors define new methods in this type of evaluating; this method is defining weight for each one of the similar or related words once appear in the student's answers.

1.2 Motivation

To enhance the accuracy of evaluation system, and its flexibility, the Fuzzy Logic "FL" in solving word's similarity problems in auto-marking system will be explored and investigated for the first time.

1.3 Problem Definitions

There are many questions should be answered before designing an auto-marking short-text answer system namely:

1. *Why there are differences in evaluating the same answer from instructor to another?*

2. *Why the instructor marking for example 3/5 and not **5/5** if the answer is correct and **0/5** if not? Why **3/5** and what is the standard points of this evaluation?*

3. *Is there any effect of spelling problems in the evaluation of the answers?*

4. *Is there any effect of grammar problems in the evaluation of the answers?*

These questions divert researchers to different problems that should be taken into consideration when develop an applications that marking or evaluating essay or short-text questions. Some of these problems are ***Syntactic Structure Analysis SSA, Rhetorical Structure Analysis RSA and Topical Content Analysis TCA.*** Which will be discussed next chapter.

 The main problem arise on surface, is that, each question has different shape of solution at the same time, there are different words having the same meaning; so the answer differ from one student to another and each answer will be most likely different from the actual answer.

Thus, marking will be different from one student to another. This problem case troubles for instructors in grading their students. Thus is the **main point** of this research. This problem likely occurs in the reading and comprehension, when student answer this question by replacing some words by synonyms "words similarity". Thus also happening when changing the words order without influencing the sentences meaning as in the following example.

What is the Relation Database?

The answer key is {a *database is a <u>set</u> of <u>related</u> <u>data</u>*}.

The student's answers may be as following:

S1: A *database is a <u>set</u> of <u>related</u> <u>data</u>*.

S2: *The database is a group of <u>data</u> that are <u>related</u> together.*

S3: *The database is a <u>set</u> of <u>data</u> that are <u>related</u> together*.

S4: A *<u>set</u> of entity that is related by foreign key*.

S5*: Is a <u>related</u> table*.

By using auto-marking (computer) all of these answers are not one hundred percent correct, except the first one, comparing by the answer key. This problem can be avoided by manually marking but it is difficult to solve this problem automatically.

Few of existing methods may solving this problem, but not in the way as the instructors need, and the main concern for them is how to build the methods as accurate as possible. Most of them fail to evaluate the short answer exactly as the instructors evaluate. This is due to the following problems:

• Some methods solved this problem by normal algorithm *"simple keyword analysis"* **[5]**. It is the simplest technique looking for the presence of coincident keywords or n-grams between the student text and the teacher one. This method cannot extract the meaning of the student answer nor deal with synonyms. The comparison results are either true or false or in another word 0/1. The differences between the key answer and the student's answer will make the comparison false or 0. So there is one possible output from this method which is 0 or 1 for each one of the main words in the answer.

• Most methods are solving this problem by making new technique rather than making normal comparison, which is dealing with similar words or synonyms as *c-rater models using Alchemist* **[14].** This method based on searching in the similar words or in synonyms database, if there is similar words in the answer, then it is true otherwise false. But this method is not accurate, since there are some similar words having different weighting than the main words.

• Some methods solving this problem by putting weight for each similar words as in Auto-Marking **[19]**. These weights defined by the instructors. Then the summation step is performed to theses weights to have the answer mark done. This method is the most effective method so far, since the results are close to the instructor's evaluation.

After the above illustration for these methods we have to introduce our algorithm. This algorithm is some how similar to the last mentioned method excepts that, we will not perform a direct summation for weights. The invented algorithm will apply a process step instead of

the summation step. This process will introduce a non-fuzzy value (word's weight) to a fuzzy value (low, med, high) with confidences, which in turn will be produce many roles to produce many fuzzy output set (zero, low, med, high, full) with confidences. These confidences are a value between zero and one.

1.4 Research Objectives

Our research objectives can be briefly described in two categories. The first identifies problems of auto-marking short-text answer, and the second to introduce a new detailed design with FL paradigm for the purpose of resolving similar or related words in student's answers text; to reduce instructor's time in educational system.

1.5 Approach and Deliverable

The word's similarity problem, in auto-marking short text answer, give us the motivation of starting with Fuzzy Logic approach to have better design. There are many software products that mark the *true -false*, multiple choice and other types of questions automatically. Few of them [ETS, e-rater, c-rater, ERB] would correct the essay-type questions (i.e. discussing, defining…etc) or short description questions.

The proposed approach for solving this problem is sited in introducing a new design auto-marking short text answer on the concept of rules as in Fuzzy Logic method.

1.6   Contributions

The main contribution of our work is how to use the Fuzzy Logic method in solving the similar words problem in short text answer. And making comparisons with other methods like keyword method and with instructor's results.

The main contributions of our work for solving the problems defined in the last section is to develop a new algorithm, **FLASA, (Fuzzy Logic based on Auto-marking Short-text Answer)**.

1.7 Structure of the thesis.

This work is organized as follows:

- *Chapter 2* starts with a brief historical overview of the field and types of questions. It also provides the importance of auto-marking essay and short-text questions, as well as a review of the current statistical, **N**atural **L**anguage **P**rocessing "**NLP**" and other techniques that are being employed and the description of the state-of-art of **C**omputer **A**ssisted **A**ssessment "CAA" for short-text answers by presenting the currently available automatic essay systems.

- *Chapter 3* presents briefly the fuzzy logic as well as describes its structural algorithm. In addition, a brief introduction of the FLASA design Methodology and it's domain.

- ***Chapter 4*** introduces our contribution with a new approach based on the use of FL algorithm. The design of original algorithm will be described, and a study of the best performance it can attain will be demonstrated.

- ***Chapters 5*** provides three examples with different number of main words, tests them, then discusses the results obtained, and makes a comparison with results obtained from other methods as well those of the instructor's evaluations. Further more, the advantages and disadvantages of FLASA will be addressed.

- ***Chapter 6*** concludes and presents the future works.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

We divide this chapter into four parts. The first part is about the concentrates on Analysis the grammatical structure of the text, and giving some tools that process it. The second section is about a history of Auto-marking system in past, recent and future research. In third parts we present *e-rater, c-rater prototype* and in four parts we present an overview of our algorithm Fuzzy Logic, we get background of this algorithm and how it works.

2.1 Introduction.

 One of the methods of monitoring individual pupil's progress is through the testing system. **Theoretically the 'test and retest**' [1] system is very effective. It can really help weaker pupils to improve on certain topics at a time. But, it also results in an increased workload for the teacher. So the technology will be used in this area to decrease workload of the instructors by making marking the answers automatically. These technologies will face problems in implements, as in our technology which is auto-marking the student's short text answers.

There are different types of questions that instructors would ask. The degrees of efficiency of these types of questions are vary. However, essay and short-text questions are very important in evaluating students for many courses, the advantages of the short-text answer over other types of questions will be discussed. More over briefly describe for the main problems in marking essay and short-text questions will be highlighted through some of algorithms and tools used.

Enable to distinguish between types of questions, a brief description for each types of questions will be studied as follow:

1.   Multiple-choice questions: This is a type of question in which students are given several possible answers and they have to choose the correct one.

a.   ○ Always Useful

b.   ○ Seldom Useful

c.   ● Depends on the situation

d.   ○ Non of the above

2.   Multiple-select questions give respondents more flexibility for giving input of feedback. In this type of questions the student can choose one or more answer.

a.   ☐ The respondent may want to choose this answer.

b.   ☑ And/or choose this answer.

c.   ☑ Or even this answer.

3.   True-false questions, are only gives an answer when the sentence or the idea is completely true or false.

○ TRUE

○ FALSE

4.   Fill in the blank ☐ can also be valuable in places where an essential single word or phrase is needed.

5.   Filter questions.

Example: (Question A is the filter question)

A. Do you like ice cream?

○   Yes (Go to question B)

◉   No (Go to question C)

B. What is your favorite ice cream flavor?

◉   Vanilla

○   Chocolate

○   Cherry Garcia

○   Other

C. Do you like cake?

◉   Yes

○   No

6. Ordinal Question

Please rank the following vegetables from best (1) to worst (5):

☐ Broccoli

☐ Carrots

☐ Potatoes

☐ Spinach

1.   Short description questions.

 Example: *What is the Fuzzy Logic?*

Fuzzy logic is a superset of conventional (Boolean) logic that
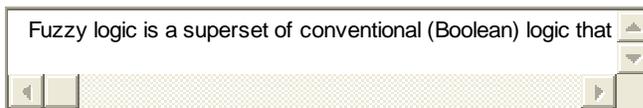
One of the main types of questions are multiple choice, True/False, Matching and Short text answer. There are advantage and disadvantage for using these types of question:

•   Multiple-choice, true/false and matching questions, unlike short-text answer questions in term of evaluating student's understanding. This is due to types of questions depends in certain way on the chance in answers. So student will answer even if could not understand the question or got the answer. These types of questions, however, can be easily evaluated without too much time consuming by the instructor. But still there are difficulty in writing some of these types especially when the questions are like multiple choice, which need more time in creating and reduce the number of questions that can be presented in the exam.

•   Short texts are perfect for brief, highly focused lessons that emphasize and reinforce teaching certain strategies or important ideas and concepts. This is because no chance available in the answer, and the students write the answer carefully with focusing. This type

of question is easy to create with limited time. In other hand, this type of question need more time than other types to be evaluated.

Although the importance of the short-text answer, there are many problems in auto-marking of this type of question, that making auto-marking very difficult to programming. So the problems and the systems that solving these problems will be discussed in next chapter.

Questioning should be used purposefully to achieve well-defined goals. An instructor should ask questions which will require students to use the thinking skills which he is trying to develop. A system exists for organizing those thinking skills. Bloom's Taxonomy is a [2] hierarchal system of ordering thinking skills from lower to higher, with the higher levels including all of the cognitive skills from the lower levels**.**

2.2 Types of questions.

The art of asking questions is one of the basic skills of good teaching. Socrates believed that knowledge and awareness were an intrinsic part of each student. Thus, **[3]** in testing the craft of good teaching an educator must reach into the student's hidden levels of knowing and awareness in order to help the student reach new levels of thinking.

Bloom **[2]** provided taxonomy for categorizing the level of abstraction of questions used in the assessment of student work. He identified six different levels that are shown in Figure 2.1. This taxonomy has been taken as the starting point for analyzing the student's learning competence. Table 2.1 summarizes the main features of each competence level.
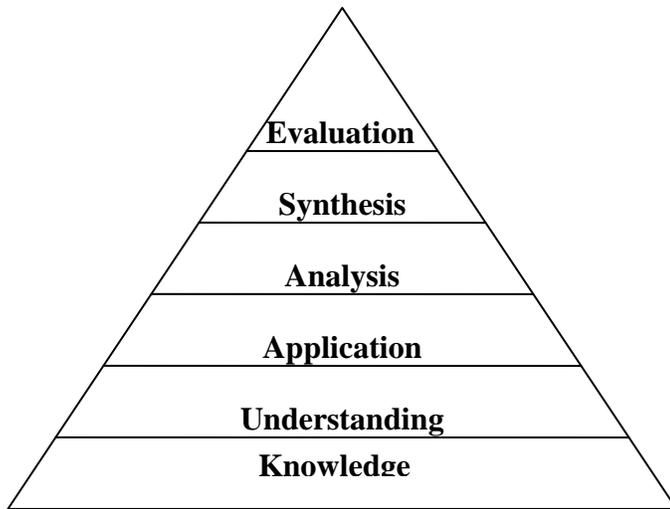
Figure 2.1: The six-element Bloom's taxonomy of educational competencies to be assessed

Table 2.1: Review of the six Bloom's competence levels, the main skill that they demonstrate, two examples of question cues and a relevant assessment method for each of them.

Competence	Question cues	Skill demonstrated	Example
Knowledge	definitions, list, concepts, principles	Remembering previously learned material	What are the stages of cell division?
Understanding	Summarize and predict	explaining in one's own words or citing examples	Explain the process of digestion?
Application	Illustrate and solve	Practical use of the material	How does the law of supply and demand explain the current increase in fruit and vegetable prices?
Analysis	Breaking a piece of material into its parts and explaining the relationship between the parts.	Notice patterns and hidden data	What is the relationship of probability to statistical analysis?
Synthesis	Putting parts together to form a new whole, pattern or structure.	Digest information	How might style of writing and the thesis of a given essay be related?
Evaluation	Using a set of criteria, established by the student or specified by the instructor, to arrive at a reasoned judgment	Judge value for purpose	How well does the Stillman Diet meet the criteria for an ideal weight reduction plan?

When it is necessary to measure the higher levels, short-text or open ended questions should be employed. Short description questions or Open ended question is the most complex type of question to evaluate because it is not a choice between several possibilities. Students have

to write a text more or less long about the topic asked with the only help of their own experience and knowledge.

The answers of an open ended question is an essay that was defined by Stalnauker in 1951 as **[5]** *"..response composed by the examinee, usually in the form of one or more sentence, of a nature that no single response or pattern of responses can be listed as correct, and the accuracy and quality of which can be judged subjectively only by one skilled or informed in the subject,...but even an expert cannot usually classify a response as categorically right or wrong. Rather, there are different degrees of quality or merit which can be recognized..."*.

2.2.1   Lower and Higher Level Questions.

Interestingly, the most instance line of experimental research on the effectiveness of questions has focused on the differences between so-called lower- and higher-level questions. Lower-level questions are generally characterized **[6]** as those that require students to recall and reiterate literally what they have read or heard. In contrast, higher-level questions are viewed as those that require more complex cognitive operations: that is, drawing information together, applying a concept, explaining, analyzing, evaluating.

The conventional wisdom seems to be that higher-level questions will stimulate deeper and more elaborate thinking that results long-lasting academic achievement than low-level questions. Even in the basis of descriptive studies, researchers who have found that teachers ask more literal questions than other question types will go on to encourage teachers to ask more high-level questions.

Usually questions at the lower levels are appropriate for:

1.   Evaluating students' preparation and comprehension.

2.   Diagnosing students' strengths and weaknesses.

3.   Reviewing and/or summarizing content.

Questions at higher levels of the taxonomy are usually most appropriate for:

1.   Encouraging students to think more deeply and critically.

2.   Problem solving.

3.   Encouraging discussions.

4.   Stimulating students to seek information on their own.

2.2.2   Open and Closed Questions

In addition to asking questions at various levels of the taxonomy an instructor might consider whether he is asking closed or open questions.

According to Bloom's Taxonomy **[2]**, a ***closed question*** is one in which there are a limited number of acceptable answers, most of which will usually be anticipated by the instructor. For example, "What is a definition for 'adjective'?" requires that students give some characteristics of adjectives and their function. While students may put the answer in their own words, correct answers will be easily judged and anticipated based on a rather limited set of characteristics and functions of adjectives.

According to Bloom's Taxonomy **[2], *an open question*** is one in which there are many acceptable answers, most of which will not be anticipated by the instructor. For example,

"What is an example of an adjective?" requires only that student's name "any adjective." The teacher may only judge an answer as incorrect if another part of speech or a totally unrelated answer is given. Although the specific answer may not be anticipated the instructor usually does have criteria for judging whether a particular answer is acceptable or unacceptable.

Both open and closed questions may be at any level of the taxonomy.

An open low-level question might be:
"What is an example of an adjective?"

An open high-level question might be:
"What are some ways we might solve the energy crisis?"

A closed low-level question:
"What are the stages of cell division?"

A closed high-level question:
"Given the medical data before you, would you say this patient is intoxicated or suffering from a diabetic reaction?"

We can close questions to finish a conversation or part of a conversation. Use open questions to get the other to speak more.

2.3 Auto-marking System

As we know that the most important thing in our life is the time, the time is very expensive in our life, and the using technology will be reducing the time of working any thing. These technologies are used in the most important things in our life. In my point of view, the most important parameter in the computer-based education and in the modern universities is that the new technology used in.

For the educational-testing community, **[7]** one motivation is economic: if you can replace two human graders with one human and one system, you can reduce the cost of grading the examination. After making case study about the time consumer in Al-Quds University for evaluating exams, the results was, 30% of instructor's time is spent on marking essay or short-text questions. As a result, instructors are not able to give writing assignment as often as they would wish. If we are to free up that 30% of their time, then we must find an effective way that teachers will trust, to mark essays and short-text responses. This substitution seems acceptable, as long as you can demonstrate that it won't affect the final grade and that human judges make the final decision, should the human and system disagree.

A second, more important, motivation **[7]** is that automated grading of short-answer questions provides students with much more immediate feedback there is no need to wait for an instructor to provide a "ruling" on the correctness of the answer. This immediacy supports interactive drills and testing, including diagnostic feedback for intelligent tutoring. However, an automated grading system's success ultimately depends on its ability to closely approximate the kinds of judgments a human grader would make.

With these problems in mind, researchers have sought to develop applications that automate essay and short text scoring and evaluation.

2.3.1 Auto-marking Limitations.

It's not easy to use computer instead of the human in evaluation system, especially when the answers are like essay or short-text answer, because these types of questions have a lot of problems and difficult in program, we will take this problems carefully in this section. All of these problems relate to Analysis the grammatical structure of the text, as will see in next section.

There are many software tools for auto-marking essay and short-text answer questions, as illustrate in figure 2.3. There have been a considerable amount of different classifications of techniques to automatically assess short-text answers. Some of them are more complete than others, but it is convenient to present at least a number of these classifications in order to fully understand the final proposed one.

In fact, any grading which involves the use of natural language, is difficult as there can be numerous ways to say the same thing. The grader has to read the student's answer and then grade it depending on how well written and how accurate the answer is. When it comes to essays, things get even more difficult, this is because **[8]** essays reflect the thoughts and opinions of another human and thus there are no perfect answers to an essay. This is indeed a very tedious and fatigue inducing job, and hence if this job is automated, we end up saving a

lot of time and money and of course, human effort. However, we believe that, in order to fully assess the answers, both a syntactic, Rhetorical and a semantic analysis is required.

- **Syntactic Structure Analysis.**

The scoring guide indicates that one feature used to evaluate an essay is syntactic variety. **[8]** The basis for syntactic analysis is parsing which is the process of making explicit the syntactic structure of sentences. This requires tagging each word in the essay with its appropriate part of speech and then assembling the words into phrases and clauses. **[9]** For example E-rater, C-rater and PS-ME are underpinned by these techniques.

- **Rhetorical "Discourse" Analysis.**

A rhetorical analyzer is used to discover the internal structure of the answer; **G**raduate **M**anagement **A**dmissions **T**est "**GMAT**" essay prompts are of two types: Analysis of an issue (issue) and Analysis of an Argument (argument). The GMAT issue essay asks the writer to respond to a general question and to provide "reasons and/or examples" to support his or her position on an issue introduced by the test question. The GMAT argument essay focuses the writer on the argument in a given piece of text, using the term argument in the sense of a rational presentation of points with the purpose of persuading the reader.

- **Lexical Conceptual Structure:**

In conformity with Oslen **[47]** this theory aims to grade essays by "fuzzy matching" of sentences that could be expressed in totally different syntactic structures but bear similar semantic content. It is based in translating the text information into language independent data structures that afterwards could be compared. The problem this approach has is very

similar to the so-called divergence problem in machine translation, when the translated text is quite different from the original one. Some solutions are proposed by Dorr **[48]**. However, none of the currently existing systems have used these ideas yet.

2.3.2 Auto-marking Techniques

There are many techniques used by most of auto-marking tools to grade essay or short-text answers, some of these techniques are:-

- **Statistical techniques:**

In general, all systems that rely on a statistical analysis of one or several features of the texts should be considered in this category. They usually need an initial training phase to calculate the parameters of the system. They do not use complex **N**atural **L**anguage **P**rocess "NLP" techniques and, in most cases, the texts are only processed with a tokens and a sentence splitter. As a consequence, they should be easy to port across languages and domains. In particular, there are several subcategories [5]:

- *Simple keyword analysis*:

It is the simplest technique and consists in looking for coincident keywords or n-grams between the student text and the teacher one. This method cannot extract a representation of the meaning of the student answer nor deal with synonyms and polysemous terms. Consequently none of the systems studied is based solely on this technique.

o *Latent Semantic Analysis LSA:* **[49]**

It is a complex statistical technique that was initially developed for indexing documents and information retrieval. Nevertheless, **[12]** it can also be applied to automated essay grading. In this field, this technique serves to extract the conceptual similarity between the student's candidate text and the teacher's reference text by looking for repeated patterns between them. According to Dessus **[50]** this approach is quite robust and proves its name by finding the hidden relationships between words that could be in different documents or between documents that do not share words. The reason for this fact is given by Landauer **[12]** who said that what causes two words to have similar meanings is that they change the meaning of passages in which they occur. He also claimed that although as many statistical techniques are language-blind, LSA might have problems with complex morphological structured languages.

- **Full natural-language processing:**

NLP is the application of computational methods to **[51]** analyze natural language. Cited tools such as syntactic parsers to find the **[52]** linguistics structure of a text and **[53]** rhetorical parsers to find the discourse structure of a text. The combination of these techniques improve the use of statistics by involving a deep text parsing and a semantic analysis in order to gather more information to effectively assess the student's answer. On the other hand, it is hard to accomplish and very difficult to port across languages. Among the current systems, C-rater and PS-ME are underpinned by these techniques. Another approach is to grade the student essay by summarizing it so that only the relevant information is taken into account and the noise is eliminated Burstein **[51]** and Marcu **[53].**

- ***Information Extraction (IE) techniques***

Information extraction (IE) techniques **[19]** pull out pertinent information from a partially syntactically analysed text by applying a set of domain specific patterns typically built from training data. Information Extraction consists in **[37]** acquiring structured information from free text, e.g. identifying Named Entities in the text and filling a template.

It can be considered as a shallow NLP technique, as it usually does not require an in-depth analysis of the texts. IE process is broken down into a series of subtasks **[19]**.

o   First, the Named Entity (**NE**) subtask consists of classifying (typically) NPs into categories like name of a person, a company, a location, or a date.

o   Secondly, relations between named entities, or properties of entities are extracted.

o   Thirdly, pre-defined templates are filled by these entities and relations.

For example, **[56]** Automark and ATM are based on this approach. An example of mark scheme is Figure 2.2.
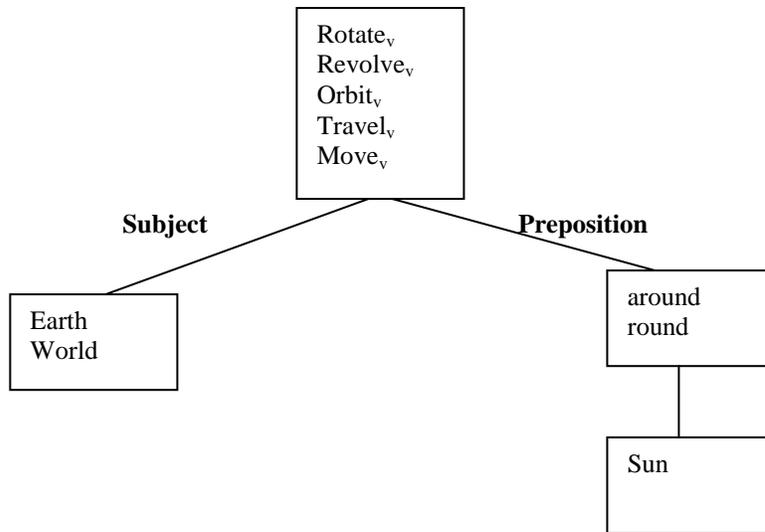
Figure 2.2: Example of a scheme used in Automark to score the answer to the question like "What movement relates the Earth and the Sun?", [56]
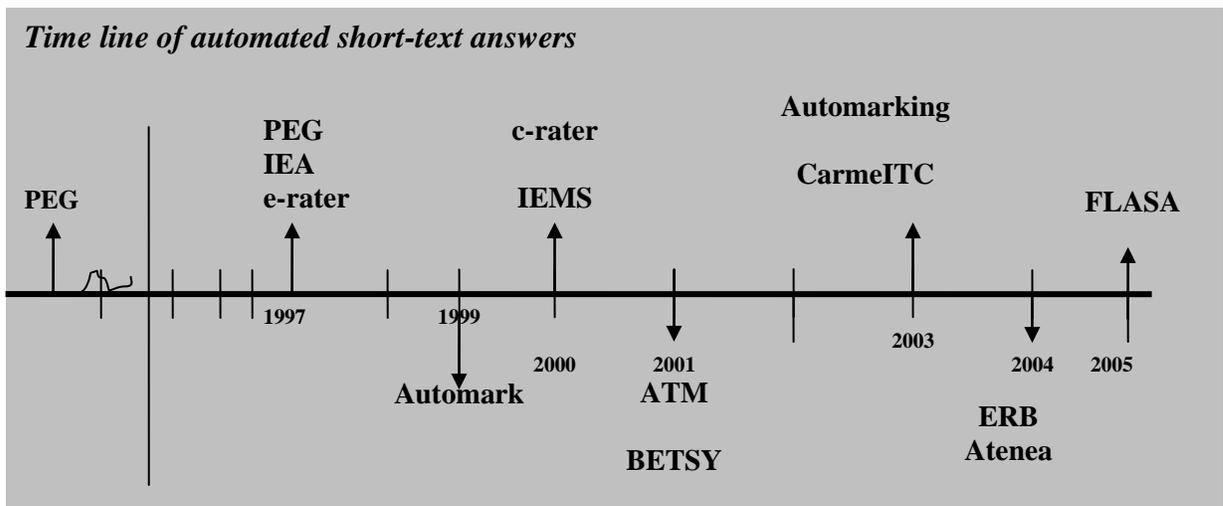
Figure 2.3: Time line of research in automated short-text answers.

2.4 Existing systems

To decrease instructor's time, researchers have sought to develop applications that automate essay scoring and evaluation. There are many methods automated essay and short-text answer questions, in this section we will introduce some of these methods, and discusses the drawback and limitation of each one.

Work in automated essay scoring began in the early 1960's and Figure 2.3 shows a timeline of automated of short questions, indicating some systems and their dates of appearance.

2.4.1 PEG "Project Essay Grader"

The pioneer in the field of auto-marking of free text answers was Page with the **P**roject **E**ssay **G**rader (**PEG**).   It focused **[7]** in the style of the essay, he set the stage for automated writing evaluation. The aim he pursued was to improve the assessment process. Despite its impressive success at predicting teachers' essay ratings, the early version of PEG received only limited acceptance in the writing and education community, precisely because it used indirect measures of writing skill. Critics argued that using indirect measures left the system vulnerable to cheating, because students could artificially enhance their scores using tricks they could simply write a longer essay.

In middle of nineties, PEG system was also undergoing transformations to include more direct measures of writing quality. In 1995, Page reported that PEG's "current programs explore complex and rich variables, such as searching each sentence for soundness of

structure and weighing these ratings across the essay."[*11*] In 1997, Page's system became mature and started to be commercially available.

PEG is **[40]** suitable for most type of essays, achieving a 87% correlation. Nevertheless, it has problems whenever the text content and word order are important, for instance, in factual disciplines' essays.

The research in this field nearly stopped until the nineties, with some exceptions, in the early of 1980s WWB Writer Work Bench tool **[38]** set took a first step toward direct measures of writing quality. (**WWB**) was not an essay-scoring system. Instead, it aimed to provide helpful feedback to writers about spelling, diction, and readability.

In 1990, the situation slightly started to change. A team of **E**ducational **T**esting **S**ervice **ETS** researchers, led by Jill Burstein, hypothesized a set of linguistic features that might more directly measure these general writing qualities-features they could automatically extract from essays using **NLP** and **I**nformation **R**etrieval **IR** techniques.

In 1993, Wresch **[54]** noted that results were not so encouraging as Page envisioned: most of the teachers did not even know of the existence of automatic software tools to assess students' essays and the research community was still exploring the field.
In addition to this, in the same year three new systems were introduced:

❖   The **I**ntelligent **E**ssay **A**ssessor (**IEA**).

❖   E-rater.

2.4.2 Intelligent Essay Assessor (IEA)

The **IEA**, **[12]** developed by Landauer and his colleagues at the Colorado University in USA in 1997. It primarily focuses on the content and it is based on Latent Semantic Analysis (LSA). LSA aims at going beneath the essay's surface vocabulary to quantify its deeper semantic content.

It was originally conceived as an academic product but some years later they founded their own company called Knowledge Analysis Technology" Website: http://www.knowledge-technologies.com" and they are in the process of patenting their system. Moreover, IEA cannot be executed in an ordinary PC but on a secure web servers placed in their company in USA. They always depend on Knowledge Analysis Technology' servers.

One of its main advantages is its language independence, with the restriction that it is not able to process too complex morphological structure of the language. Also it does not use any NLP techniques such as removing stop words.

It is also possible to perform synonym recognition in order to treat several synonyms with similar meanings as the same word. IEA requires an initial training but it is not human supervised. The only input is a set of texts about the topic to evaluate.

IEA has been tested in the military environment with 2000-word essays and achieving a 0.35 interreliability between the teacher and the system. IEA has also been used for psychology, medicine and history texts, achieving a 80%-90% of exact agreement with the teacher.

Landauer et al. stated that one problem their system has is that it does not take into account the word order. Thus, it cannot interpret sentences in which word order is the discriminate factor. Besides, it is easily tricked because it does not perform any syntactical or grammatical analysis.

According to its authors, IEA can be used in many different applications within education, from the simple consistency checker, to help teachers to discover cheating and plagiarism, to the formative and summative assessment of the essays.

2.4.3   E-rater.

E-rater, an improved version of **E**ducational **T**esting **S**ervice "**ETS**" which uses a hybrid approach by combining NLP and statistical techniques. It was presented in 1997 **[39]** and in 1999 it became the second grader for the GMAT exam in USA. E-rater features are based on four general types of analysis: syntactic, discourse, topical, and lexical, so It takes into account both the content and the style of the text.

A diagram of the E-rater architecture can be seen in Figure 2.4. E-rater relies on the combination of statistical techniques and NLP. It takes into account both the content and the style of the text. The content is checked by a vector of weighted content words and the style with shallow parsing techniques to identify syntactic and discourse features.
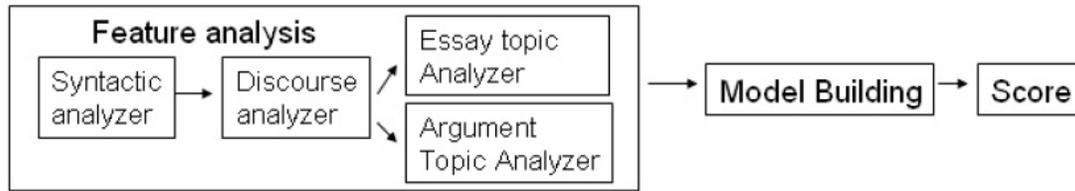
Figure 2.4: Architecture of the E-rater system **[39]**

It is important to notice that in some cases E-rater **[51]** is not able to score the text because it is too short, or too different from the rest. In these cases, an advisory message is generated.

Since 1999, **[5]** E-rater has scored over 750,000 GMAT essays with an agreement about 97% with the other grader.

The main problems that have been found is that **[40]** the system could be tricked by writing grammatically correct texts but without meaning, also indicate that E-rater does not assess text content beyond keywords identification and that it cannot deal with essays on factual disciplines.

2.4.4 Automark.

In 1999 **[18]** Mitchell, Russell, Broomhead and Aldridge from the University of Liverpool and Brunel University in UK created a new automated methods called *Automark*. The aim of the system is mostly summative, that is, to grade the style and the content of a student essay in order to say whether it is acceptable or not according to the criteria specified by the teacher to the system.

AutoMark uses **[42]** IE techniques and some NLP techniques to ignore some mistakes in spelling, typing, syntax or semantics that should not be taken into account.

The system has been used in the **[18]** Brunel University to test Java knowledge of first year engineering students, and it has also been applied to assess answers from the 1999 statutory national curriculum assessment of science. In this case, students were 11-year-old pupils, and there were four types of questions: single word generation, single value generation, generation of a short explanatory sentence and description of a pattern in data.

The correlation achieved ranged between 93% and 96%. Finally, four problems can be identified: to correctly identify misspelled words, to correctly analyze the sentence structure, to identify an incorrect answer, and to assess information that is not represented in the mark scheme template. The distribution of each problem is shown in the Figure 2.5.
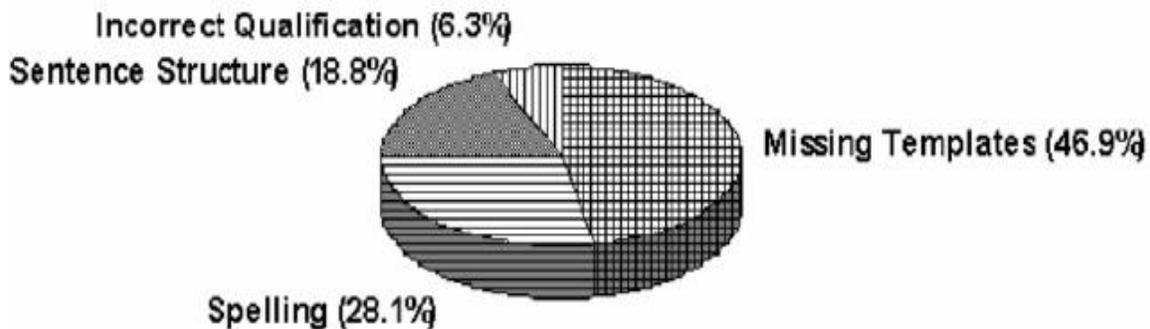
Figure 2.5: Drawbacks in the assessing process of AutoMark, source Mitchell et al. (2002)**[43]**

49

2.4.5   C-rater

In the year 2001 a prototype of an automated scoring engine called Concept-rater or *C*-rater, has been developed at the Educational Testing Service (ETS), to measure a student's understanding of specific content material without regard for the student's writing skills. It uses [5] automated NLP techniques to determine whether a student response contains specific linguistic information required as evidence that the concept has been learned.

It is [14] currently a working system and the scoring process is fully automatized, but the reference model-building still requires human intervention. Hence, they have developed Alchemist, a friendly interface for this task.

C-rater [14] tries to recognize when a response is equivalent to a correct answer, and so is, in essence, a *paraphrase recognizer*. As such, the scoring engine is designed to recognize a correct response when it exhibits the variations that are ordinarily associated with paraphrases, whether they be syntactic variation, different inflections of a word, substitution of synonyms or similar terms, or the use of pronouns in the place of nouns.

In addition to these features, which are ordinarily associated with paraphrasing, c-rater recognizes words that are spelled incorrectly – an essential feature for the K-12 market. Table 2.2 shows examples of these paraphrase variations as they have appeared in student responses.

Table 2.2: Types of Variation in C-rater Method [14]

Syntactic Variation	**Money worries Walter → Walter is worried about money.**
Inflectional Variation	**dreams, dreaming →dream**
Synonymy or Similarity	**dreams →wants** **expensive →costly**
Pronoun Reference	**Mama disagrees with Walter. He thinks that money is life.**
Spelling	**Walter →Wlater, Waalter, Walther**

According to Diana **[5]**, this system is very similar to the E-rater system. In fact, their main differences are that E-rater focuses on the style, while C-rater on the content; that E-rater assigns a holistic score, while C-rater only identifies whether the response contains specific information necessary to be correct; that E-rater is partly based on the rhetorical structure of an essay, while C-rater is more based on a predicate-argument structure; and that E-rater needs a larger training set.

C-rater has been usually applied to formative low-stakes tasks, as for example the review short questions at the end of each chapter in a textbook. According to Diana **[5]** when C-rater was used in a small-scale study with a university virtual learning program it achieved over 80% of agreement with the instructor and, according to Leacock **[41]**, when it was used in a large-scale assessment to score 170,000 short-answer responses to 19 reading comprehension and five algebra questions, the result was 85% of accuracy.

In compliance with Leacock **[41]**, C-rater's main problems are that it is unable to assess answers that depend on the verb tense due to the stemming phase, that it does not deal with answers that include a quote, that some spelling mistakes are not correctly repaired, and that it does not know how to manage idiomatic expressions.

2.4.6 IEMS Intelligent Essay Marking System

The Intelligent Essay Marking System was presented by **[44]** Ming, Mikhailov and Kuan from the NGEE ANN Polytechnic in Singapore, in 2000. Its aim is both summative and formative.

IEMS is based on the Pattern Indexing Neural Network, the Indextron that performs pattern recognition and in this case the patterns are the words of the texts. This system has been mostly applied to qualitative questions (e.g. biology, psychology, history or anatomy) rather than numerical ones. For instance, taking an 800-word passage entitled "Crime in Cyberspace" and asking 85 students of third-year Mechanical Engineering to write a summary of not more than 180 words about the text, IEMS achieved an 80% correlation.

2.4.7 ATM Automated Text Marker.

In the year 2001 **[15]** at the Portsmouth University in the UK Callear, Jerrams-Smith and Soh developed a new automated method called **A**utomated **T**ext **M**arker **ATM**. They were so convinced that both content and style should be taken into account that they designed their system in order to give two independent score, one for each aspect and to leave the teacher the task of combining them to give the final grade.

ATM relies on IE techniques to assess students' essays. The system architecture is shown in Figure 2.6. It is important to highlight the syntax and semantics analyzer:

• *The syntax analyzer*: It checks the grammar of each input sentence. According to Callear [45], this can be done successfully.

• *The semantics analyzer:* The system [45] looks for concepts in the text and their dependencies, and then a pattern-matching Prolog procedure is performed between the dependency groups from the student's answer and the reference model. See Figure 2.7 for an example of dependencies group.
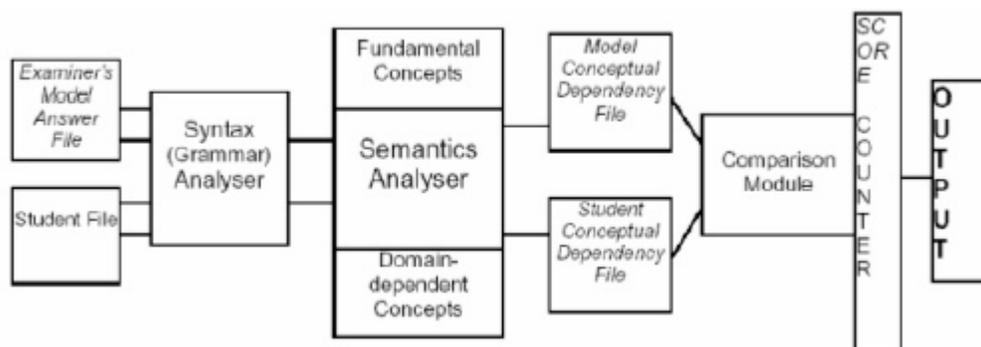
Figure 2.6: Architecture of the ATM system, source Callear et al. (2001) [45].

According to its authors, ATM works better assessing short answers to factual questions (e.g. in Prolog programming, psychology and biology-related fields).
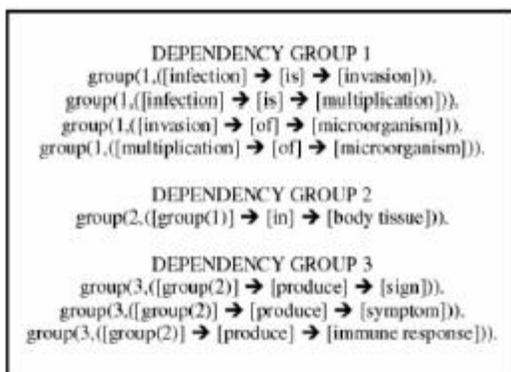
Figure 2.7: Example of dependencies groups found out by the semantics analyzer of ATM, source Calleat et al. (2001) [45].

2.4.8   BETSY "Bayesian Essay Test Scoring sYstem".

The *BETSY* system was [16] developed between 2001 and 2003 by Rudner and Liang at the College Park of the University of Maryland with funds from the U.S. Department of Education. According to author, its aim is to classify essays using a four point nominal scale.

BETSY is underpinned by naive bayesian networks. The user is given the possibility of choosing one of two models: Multivariate Bernouilli Model (MBM) and Bernouilli Model (BM). Rudner and Liang claim that BM is quicker as it only looks if certain features are present while MBM takes into account the uses in which these features have been employed. A comparison between both models is done by McCallum, A. and Nigam [17] and they suggest that MBM with a large vocabulary is more accurate than BM. Although as Rudner and Liang warn it might be different with students' essays.

BETSY has the [18] possibility of stemming the text and removing the stop words, this might improve the text classification task. The system has been used to assess Biology items for the

Maryland High School and the results were that the BM model achieved an 80% accuracy and the MBM a 74%. Furthermore, Rudner and Liang say that their system could be applied to any text classification task.

2.4.9   Auto-Marking.

Auto-marking was developed by **[19]** Pulman, Sukkarieh and Nicholas Raikes "*Computational Linguistics Group"* in Oxford and in the Interactive Technologies in Assessment and Learning (**ITAL**) Unit of the University of Cambridge Local Examinations Syndicate (**UCLES3**). Its aim is not to automatically score high-stakes exams, but to help in low-stakes ones. Each exercise is given a value between 0 and 2, where 0 means incorrect, 1 partially correct or incomplete, and 2 correct and complete.

This system relies on a combination of NLP and pattern-matching techniques. It consists of three modules:

• *Customization and shallow processing module*: Firstly, **[19]** it uses a Hidden Markov Model part-of-speech "HMM POS" tagger, and a Noun Phrase (**NP**) and Verb Group (**VG**) Finite State Machine (**FSM**) chunker to provide the input to the information extraction pattern matching modules as in figure 2.8.

• *The pattern-matcher module*: It is very similar to the one used in Automark, that is, human experts have to design the information extraction patterns and then the students' answers are

compared against them. Appelt and Israel **[20]** emphasized the importance of designing good rules.

The 3 crucial steps in which to write extraction rules by hand can be found **[20]**, these, in order, are:

1.   Determine all the ways in which the target information is expressed in a given corpus.

2.   Think of all the plausible variants of these ways.

3.   Write appropriate patterns for those ways.

• *The marking algorithm module*: These rules are organized in classes and the algorithm described in **[19]** matches them with the student's processed answer to score it.

AutoMark has been applied with answers from the GCSE exam of Biology with a 88% of exact agreement between the teacher and the system. On the other hand, the authors claimed that this system is not suitable for subjective general opinions and therefore it should not be used in that area.

The main problem they encountered was the **[5]** inaccuracy of taggers that do not have enough knowledge about Biology, even that they include some guessing heuristics for unknown words. Moreover, the system cannot deal with students' inferences and with contradictory or inconsistent information.

<div style="text-align:center">

Biology Text

"When the caterpillars are feeding on the tomato plants, a chemical is released from the plants."

</div>

When/WRB [the/DT caterpillars/NNS]/NP [are/VBP feeding/VBG]/VG on/IN [the/DT tomato/JJ plants/NNS]/NP,/, [a/DT chemical/NN]/NP [is/VBZ released/VBN]/VG from/IN [the/DT plants/NNS]/NP./.

<div style="text-align:center">
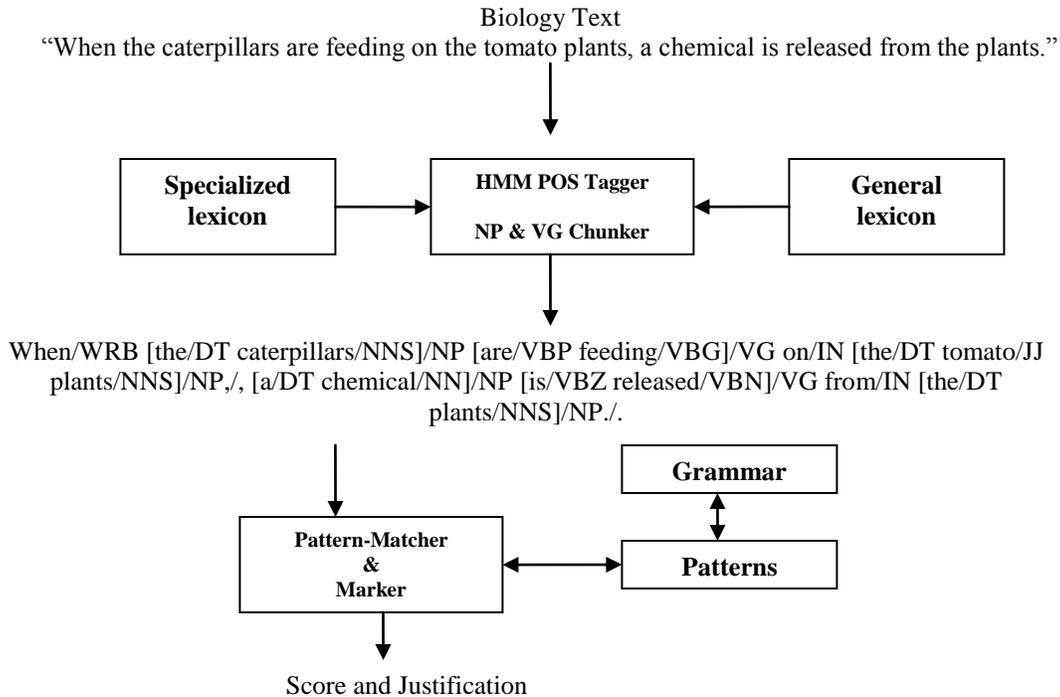
Figure 2.8: Auto-marking Modules [19]

</div>

2.4.10 CarmelTC

Carmel is a Virtual Learning Environment "**VLE**" system that has been recently incorporated a new free text assessment module called CarmelTC. This module has been developed at the University of Pittsburgh by Ros´e, Roque, Bhembe and Vanlehn **[34]**. CarmelTC has also been used in the tutorial dialogue system Why2.

CarmelTC relies on the combination of machine learning classification methods using the features extracted from the Carmel's linguistic analysis of the text and the Rainbow Naive Bayes classification **[21]** .The system was tested with 126 physics essays, and the results were 90% of precisions, 80% of recall and a 8% of false alarm rate.

<div style="text-align:center">

57

</div>

2.4.11 ERB "Evaluating Responses with B_{LEU} "

It's a new automated evaluation methods based on the use of a shallow modified version of the **Bleu** "**Bi**Lingual **E**valuation **U**nderstudy" algorithm, which is a translation system presented by Papineni, Roukos, **[22]**, whose main goal is to rank systems according how well they translate the texts from one language to other.

Due to the good results attained by the Bleu algorithm, some researchers started to think about applying this algorithm to new fields. For instance, Lin and Hovy **[10]** used it for evaluating summaries. It depends on the references text to evaluate candidate's text. Including shallow NLP techniques, such as stop words removing and stemming and by removing the Brevity Penalty BP factor, as for summaries the idea is just the opposite.

There are three main types of questions have been assessed with ERB: definitions, advantages/disadvantages and yes/no with justification, the performance of evaluating these types of questions are shoed in figure 2.9. The crucial factors affecting the ERB performance are the number and quality of the reference texts used **[5]**.

The problem of simple using of the Bleu algorithm to evaluate students' answers was that it only takes into account the precision and ignores the recall, that is, it does not penalizes students' texts that do not cover some percentage of the information in the reference texts. Therefore, they have modified the **[22]** BP factor in order to consider the recall too and we have called this new BP factor, the Modified Brevity Penalty (MBP) factor. The core idea of this MBP factor is shown in Figure 2.10.
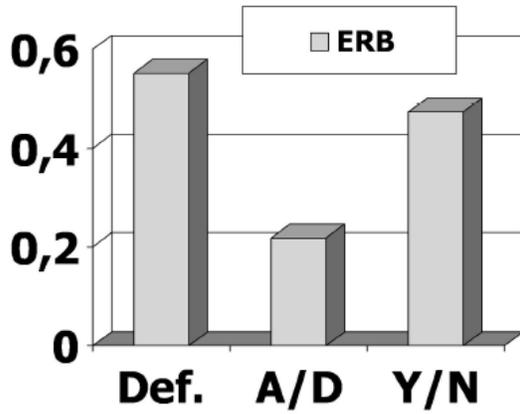
Figure 2.9: Histogram that shows how different types of questions affect ERB performance. Def. stands for definition or description, A/D for advantage/disadvantage and Y/N for yes/no with justification.
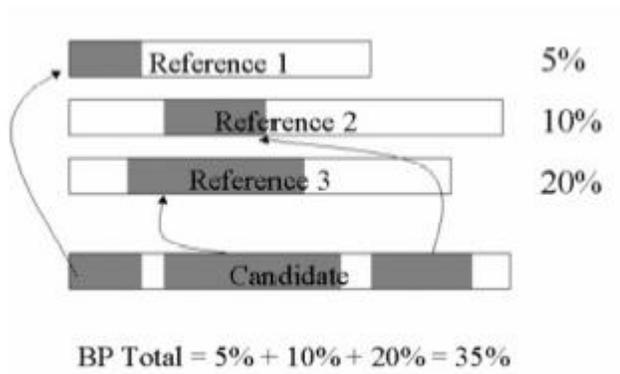
Figure 2.10: Graphical visualization of the procedure to compute the MBP factor.

CHAPTER 3

FUZZY LOGIC AND METHODOLOGY

3.1 Fuzzy Logic.

Most of us have had some contact with conventional logic at some point in our lives. In conventional logic, **[23]** a statement is either true or false, with nothing in between. This principle of true or false was formulated by Aristotle some 2000 years ago as the Law of the Excluded Middle, and has dominated Western logic ever since.

As the complexity of a system increases **[*30*]**, it becomes more difficult and eventually impossible to make a precise statement about its behavior, eventually arriving at a point of complexity where the fuzzy logic method born in humans is the only way to get at the problem.

The term "fuzzy logic" emerged in the development of the theory of fuzzy sets by Lotfi Zadeh **[24]** a professor at the University of California at Berkley in (1965) when he presented his seminal paper on "fuzzy sets. Zadeh showed that fuzzy logic unlike classical logic can realize values between false (0) and true (1).

Basically, he transformed the crisp set into the continuous set [1,2], in other meaning, there are different possible values between 0 and 1 in the crisp set of [0, 1], 0 for absolutely false and 1 for absolutely true. Computers can interpret only true or false values but a human being can reason the degree of truth or degree of falseness.

Fuzzy models interpret the human actions and are also called intelligent systems **[24]**, as in the Fuzzy sets **[23]** which have movable boundaries, *i.e.*, the elements of such sets not only represent true or false values but also represent the degree of truth or degree of falseness for each input.

FL was conceived as a better method for solving many types of "real-world" problems, especially where a system is difficult to model, **[25]** it has proven to be an excellent choice for many control system applications since it mimics human control logic. **[26]** It uses an imprecise but very descriptive language to deal with input data more like a human operator.

Some Fuzzy Logic applications include:

- Control (Robotics, Automation, Tracking, Consumer Electronics)

- Information Systems (DBMS, Info. Retrieval, Data Mining)

- Pattern Recognition (Image Processing, Machine Vision)

- Decision Support (Adaptive HMI, Sensor Fusion) .

According the concepts of fuzzy logic, a typical fuzzy system consists of a rule base, membership functions, and an inference procedure as in Figure 3.1.

| Input → | Crisp to Fuzzy **Fuzzify** | → | Inference Max, Min .etc | → | Fuzzy to crisp Defuzzify | Output → |
| Fuzzy Set Membership Functions | | Rule Base | | Max , average, centroid ,singleton , etc | |
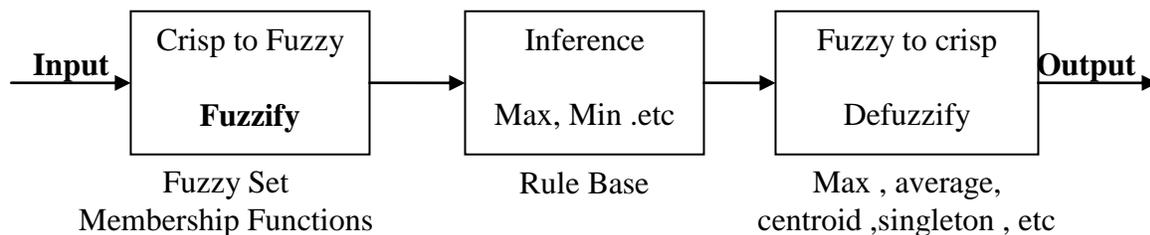
Figure 3.1: Scenario of FL System. [25]

3.1.1 Fuzzification

In any fuzzy logic algorithm it must be define input and output fuzzy set, ***Defining the input and output data is perhaps the most important step of all in constructing an expert system.*** Input fuzzy set for replaced the input crisp number into any linguistic words from input fuzzy set members, and the output fuzzy set.

Establishes the fact base of the fuzzy system. First, **[27]** it identifies the input and output of the system "Fuzzification", then defines appropriate IF THEN rules and uses raw data to derive a membership function. At this point, one is ready to apply fuzzy logic to the system.

The fuzzification methods is one of the important steps of any fuzzy algorithm, in this step we converting a crisp input number into linguistics words from fuzzy set as we will define. In this step we will define a term of Fuzzy set, Fuzzy subset and the membership functions.

3.1.1.1 Classical Set

In mathematics as we taken in set theory course, a set is simply a collection of objects "numbers, characters or strings", which have a common trait. One may associate with every crisp set S, a membership value μ_s: U $\rightarrow$ {1, 0} to every element in the universe of discourse U. A value of 0 is assigned if the element does not belong to the set and a value of 1 is assigned if the element belongs to the set **[28]**.

For example if we have a crisp set X of all real number between 0 and 1. From this set X a subset A can be defined, (e.g. all values $0 < g < 0.2$). The *characteristic function* of A, (i.e.

this function assigns a number 1 or 0 to each element in *X,* depending on whether the element is in the subset *A* or not) is shown in Fig.3.2 .**[29]**
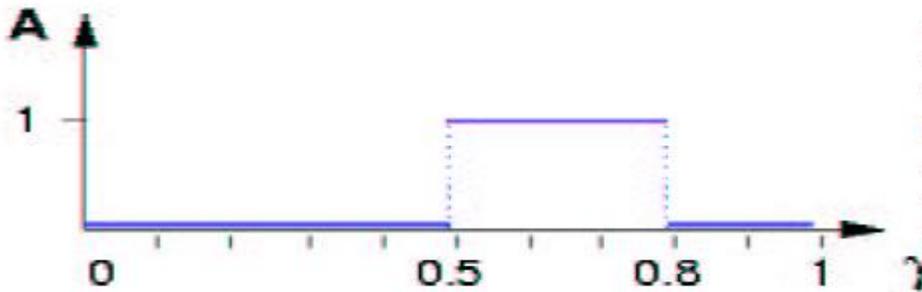
Figure 3.2. Characteristic Function of a Crisp Set [29].

Any elements belong to the subset A are assigned by the number 1 and any elements are not belongs to the subset A are assigned to the number 0.This concept is restricted, and lacks of flexibility for some applications like controls system, information system, Decision Support System… etc.

3.1.1.2 Fuzzy Set

Humans do not think in binary terms. In general, we realize that things occur in degrees; fuzzy sets enable computers to map this way of thinking. Quality is not either high or low only; for example, we may describe it as very high, high, adequate, low, and very low. An objective of fuzzy sets is to make computers think like humans.

The aim is to use fuzzy sets in order to make computers more intelligent; therefore, the idea above has to be coded more formally. In the example, all the elements were coded with 0 or 1. A straight way to generalize this concept is to allow more values between 0 and 1. In fact

infinitely many alternatives can be allowed between 0 and 1, namely the unit interval I = [0, 1] **[29]**.

A fuzzy set as Lotfi Zadeh proposed **[30]**, is one to which objects can belong to different degrees, called *grades of membership* in other way fussy set is a group of anything that cannot be precisely defined. Is described using membership function, $\mu_F: U \rightarrow [1, 0]$. Each element x in the universe of discourse U is assigned a degree of membership $\mu_F(x)$ in [0, 1].

To illustrate the idea of fuzzy sets, suppose we want to classify the room temperature, we have four classifications of a temperature. *Very cold, cold, normal* and *hot*. Firstly we want to describe our classification by using classical set as in figure 2.13 here we have a crisp set X for real number between 0 to 40 , from this set we have four subset , *very cold*, cold, *normal* and *hot*, *very cold* defined in subset {0,10} , *cold* →{10,20} , *normal*→ {20,30} and *hot* → {30,40}.

If we want to classify a room temperature which is 15, from all subsets we defined this temperature belongs to the subset *cold*, so it's 1 for subset *cold* and 0 for other subsets as in figure 3.4 . What about 19.99 or 30.00099, if we want to implements classical set methodology we have that 19.99 belongs to subset *cold*, and 30.00099 belong to subset D.

3.1.1.3 Fuzzy Subset

The label *cold* can be translated to a fuzzy set which is temperature of room with every temperature associated with a value from zero (no cold) to one (cold) to represent the degree of cold temperature as we think about it. We can then write the fuzzy subset cold:

Cold = {(5,0), (10,0.66), (12.5,1), (15,0.66), (17.5,0.33), (20,0)}   (1)

$$\mu_{\text{very cold}}(X) = \begin{cases} 1 & \textbf{IFF } X <= 10 \\ 0 & \textbf{IFF } X > 10 \end{cases}$$

$$\mu_{\text{cold}}(X) = \begin{cases} 1 & \textbf{IFF } 10 < X <= 20 \\ 0 & \textbf{IFF } X > 20 \end{cases}$$

$$\mu_{\text{normal}}(X) = \begin{cases} 1 & \textbf{IFF } 20 < X <= 30 \\ 0 & \textbf{IFF } X > 30 \end{cases}$$

$$\mu_{\text{hot}}(X) = \begin{cases} 1 & \textbf{IFF } 30 < X \\ 0 & \textbf{IFF } X < 30 \end{cases}$$

Figure 3.3: Classical set membership functions for the room temperature.
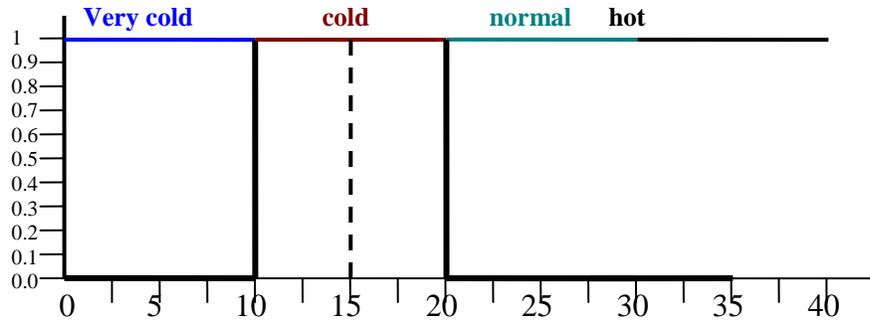
Figure 3.4: Classical set for room temperature classification.

The fuzzy subset defined by (1) reflect the way of thinking that *room Temperature* 12.5 lead to completely cold *Temperature*, 10,15, and 17.5 are somewhat cold *Temperature*, and 20, 5 are completely not cold *Temperature* as in figure 3.5 which describe the membership functions for each one of the fuzzy subset for the room *Temperature* set.
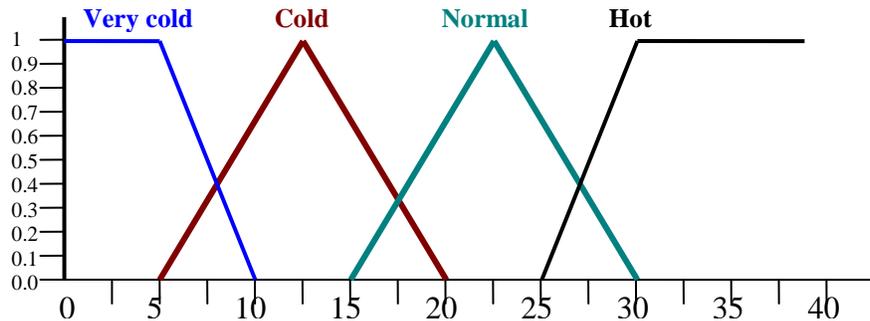
Figure 3.5: .membership functions for Fuzzy set for room Temperature classification.

As we see each one of fuzzy set members is a subset of main set which is room *Temperature*. From this example we can define terms *fuzzy subset* as, a *fuzzy subset* F of a fuzzy set S, is a set of order's pairs, first elements is an element of the fuzzy set S " room *Temperature*", and

66

a second element that is the value of interval [0,1] " the grade of the room *Temperature* for this subset F".

3.1.1.4 Fuzzy Variable

Several fuzzy sets representing linguistic concepts such as low, medium, high, and so on are often employed to define states of a variable. Such a variable is usually called a *fuzzy variable [31 page 13-14]*. Accordingly, linguistic variables are a critical aspect of some fuzzy logic applications, where general terms such a "very cold" "cold" "normal" and "hot" are each used to capture a range of numerical values, as in figure 3.3. While similar to conventional quantization, fuzzy logic allows these stratified sets to overlap (e.g., a 17.5 room *Temperature* may be classified in both the "cold" and "normal" categories, with varying degrees of belonging or membership to each group). Since these fuzzy variables are more attuned to reality than crisp variables.

3.1.1.5 Input and output fuzzy set

In any fuzzy system we have to define the input fuzzy set which is the classification of things as we described, and output fuzzy set which describe the system actions, For example, an extremely simple temperature regulator that uses a fan, which has input fuzzy set or input fuzzy variable {very cold, cold, hot, very hot}, and output fuzzy set or output variable {stop fan speed, turn down fan speed, maintain level, speed up fan}.

After we define the fuzzy set of the fuzzy system, We need to describe a membership function for each one of the input fuzzy set members, to describe the members behavior, this because each input may be have to be replaced by one or two words, but which words, the

membership function can be describe which words can be replaced with confidence for each linguistic words.

Note: we will use confidence rather than degree of the membership function of fuzzy set members. That because we may use terms degree to indicate the student's answer degree or marks later in this paper, and this permits us to use ordinary language in describing things in a precise way.

3.1.1.6 Membership function.

[31 page 11] Each fuzzy set is completely and uniquely defined by one particular membership function; the membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. [26] There are different memberships functions associated with each input and output response.

The membership function of a fuzzy set F is denoted by $\mu_F(x) : U \rightarrow [0,1]$, describes the membership of the elements x of the fuzzy set F in U, whereby for μ_F a large class of functions can be taken.

The grade or confidence of membership $\mu_F(X)$ of a membership function $\mu_F(X_0)$ describes for the special element $X=X_0$, to which grade it belongs to the fuzzy set F. This value is in the unit interval [0, 1]. Of course, X_0 can simultaneously belong to another fuzzy set F_1, such that $\mu_{F1}(X_0)$ characterizes the grade of membership of X_0 toF_1.

There are a membership function for both input and output fuzzy set, the input membership function for converting input number to linguistic words "fuzzification", and output membership function for converting linguistic words to crisp number "defuzzification".
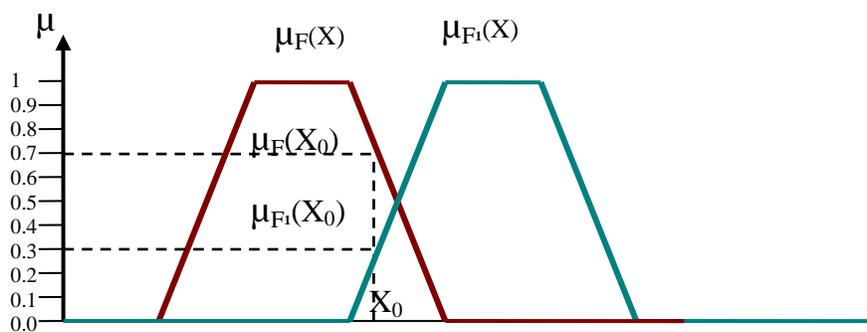
Figure3.6: .membership functions for Fuzzy set $\mu_F(X)$ and $\mu_{F_1}(X)$

There are different shapes for the membership function, like trapezoidal and triangular member ship function, trapezoidal as in figure 3.6, and triangular as in figure 3.5.

3.1.1.7 Operators for fuzzy sets

The basic connective operations in classical set theory are those of *intersection*, *union* and *complement*. These operations on characteristic functions can be generalized to fuzzy sets in more than one way. However, one particular generalization, which results in operations that are usually referred to us as standard fuzzy set operations, has a special significance in fuzzy set theory. In the following, only the standard operations are introduced. The following operations can be defined:

- The *fuzzy intersection* operator ∩(fuzzy AND connective) applied to two fuzzy sets A and B with the membership functions $\mu_A(X)$ and $\mu_B(X)$ is

$$\mu_{A \cap B}(X) = \min \{ \mu_A(X) , \mu_B(X) \}$$

- The *fuzzy union* operator ∪(fuzzy OR connective) applied to two fuzzy sets A and B with the membership functions $\mu_A(X)$ and $\mu_B(X)$ is.

$$\mu_{A \cup B}(X) = \max \{ \mu_A(X) , \mu_B(X) \}$$

- The *fuzzy complement* (fuzzy NOT operation) applied to the fuzzy set A with the membership function $\mu_A(X)$ is

$$\mu_{\bar{A}}(X) = 1 - \mu_A(X)$$

3.1.2   Fuzzy Rule Base "Fuzzy Inference"

In the last sections we describe fuzzy set, and define the input and output fuzzy variable, in this section we have to define the fuzzy rule base, these rules are mapping between inputs fuzzy set and outputs fuzzy set.

Fuzzy rule-based approach to modeling is based on verbally formulated rules overlapped throughout the parameter space. They use **[32]** numerical interpolation to handle complex non-linear relationships. It's linguistic IF-THEN- constructions that have the general form "IF A THEN B" where A and B are (collections of) propositions containing linguistic variables. A is called the *premise* and B is the *consequence* of the rule.

In effect, the use of linguistic variables and fuzzy IF-THEN- rules exploits the tolerance for imprecision and uncertainty. In this respect, fuzzy logic mimics the crucial ability of the human mind to summarize data and focus on decision-relevant information.

For example:

IF **temperature** IS **very cold** THEN **stop fan**

IF **temperature** IS **cold** THEN **turn down fan**

IF **temperature** IS **normal** THEN **maintain level**

IF **temperature** IS **hot** THEN **speed up fan**

Notice there is no "ELSE". All of the rules are evaluated, because the temperature might be "cold" and "normal" at the same time to differing degrees.

Here we have linguistics variable for input fuzzy set which is temperature classification {cold, very cold, normal, hot}, and the output fuzzy set which is the fan speed action {stop, turn down, maintain level, speed up}. We have in this example 4 rule bases. The number of rules depends on number of inputs and number of members in the input fuzzy set. The number of rules is calculated by the following function:

$$\textbf{Number of rules} = \textbf{Z}^{\textbf{n}} \qquad \textbf{(2)}$$

Where Z is the number of members in the input fuzzy set which is 4 members in room temperature example, and n number of inputs which is the room temperature and its crisp number. So we can derived a fuzzy rule base format from equation (2).

$$\textbf{If } \textbf{I}_1 \textbf{ is f}_{C1} \textbf{ and } \textbf{I}_2 \textbf{ is f}_{C2} \textbf{ and } \dots. \textbf{ In is f}_{Cn} \textbf{ then output is O}_{min(C1,C2\dots,Cn)} \qquad \textbf{(3)}$$

Where I is the input value, n is the number of inputs, f one of input fuzzy set members , C is the confidence of each one of the members of the fuzzy set," each one of the members of fuzzy set has its own membership function as in *figure 3.5,* from these membership functions we can calculated the confidences of each one of the members for each rule defined in this part", and O is one of the output fuzzy set members, and $\min(C_1, C_2..Cn)$ is the results of AND operation as we discuses in last section.

As we said in last section, there is a unique membership function associated with each input parameter. The membership functions associate a weighting factor "**Confidence**" with values of each input and the effective rules. These weighting factors determine the degree of influence or degree of membership (**DOM**) each active rule has. By computing the logical product of the membership weights for each active rule, a set of fuzzy output response magnitudes are produced. All that remains is to combine and defuzzify these output responses.

3.1.3 Defuzzification

Fuzzy logic **[23]** is a rule-based system written in the form of horn clauses (*i.e.*, if-then rules). These rules are stored in the knowledge base of the system. The input to the fuzzy system is a scalar value that is fuzzified. The set of rules is applied to the fuzzified input. The output of each rule is fuzzy. These fuzzy outputs need to be converted into a scalar output quantity so that the nature of the action to be performed can be determined by the system. The process of converting the fuzzy output is called defuzzification.

Defuzzification is the reverse process of fuzzification. We have confidences in a fuzzy set of word descriptors as we define in our rules base, and we wish to convert these into a real number. This may be necessary if we wish to output a number to the user.

The defuzzification method is the final phase of fuzzy system, as we illustrates in figure 2.9. The purpose of defuzzification is to convert each conclusion obtained by the inference engine "rule base", which is expressed in terms of a fuzzy set, to a single real number.

3.1.3.1 Defuzzification Methods.

In the fuzzy models, [33] there are several methods of determining the expected value of the solution fuzzy region. These are the methods of decomposition, also called method of defuzzification, and describe the way we can derive an expected value for the final fuzzy state space.

There are many defuzzification techniques [24] but primarily only three of them are in common use. These defuzzification techniques are discussed below in detail.

- **Maximum Defuzzification Technique**

This method gives the output with the highest membership function. This defuzzification technique [23] is very fast but is only accurate for peaked output. This technique is given by algebraic expression as

$$\mu_A(x^*) \geq \mu_A(x) \qquad \text{For all } x \text{ in } X \qquad (4)$$

where x^* is the defuzzified value. This is shown graphically in Figure 3.7.
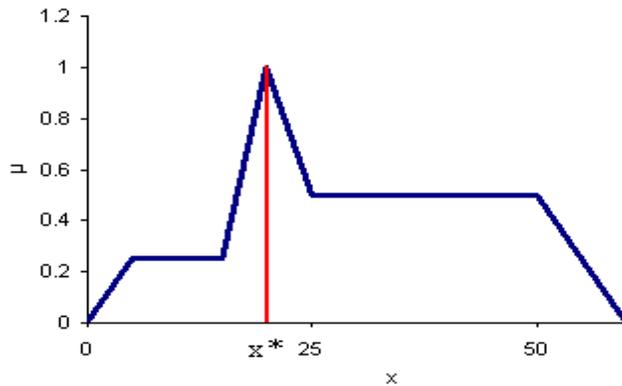
Figure 3.7: Max-membership defuzzification method

- **Centroid Defuzzification Technique**

This method is also known as center of gravity or center of area defuzzification. This technique was developed by Sugeno in 1985 **[23]**. This is the most commonly used technique and is very accurate. The centroid defuzzification technique can be expressed as the weighted strengths of each output member function are multiplied by their respective output membership function center points and summed. Finally, this area is divided by the sum of the weighted member function strengths and the result is taken as the crisp output.

$$x^* = \frac{\int \mu_i(x)\, x\, dx}{\int \mu_i(x)\, dx}$$

Where x^* is the defuzzified output, $\mu_i(x)$ is the aggregated membership function and x is the output variable. The only disadvantage of this method is that it is computationally difficult for complex membership functions.

74

- **Weighted Average Defuzzification Technique**

In this method the output is obtained by the weighted average of the each output of the set of rules stored in the knowledge base of the system. The weighted average defuzzification technique can be expressed as

$$x^* = \frac{\sum\limits_{i=1}^{n} m^i w_i}{\sum\limits_{i=1}^{n} m^i}$$

Where x^* is the defuzzified output, m^i is the membership of the output of each rule, and w_i is the weight associated with each rule. This method is computationally faster and easier and gives fairly accurate result.

3.2 FLASA Design Methodology

The development of FLASA is passing through the following steps as shown in figure 3.8.
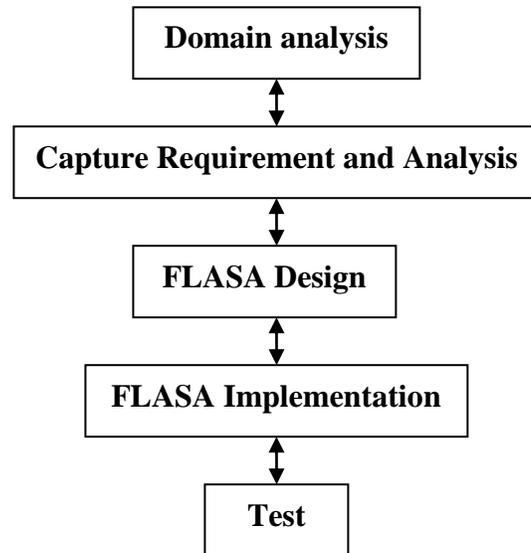
Figure 3.8. FLASA Development phase**.**

3.2.1 The Scope of Domain.

The aim of the proposed **FLASA** is to generate auto-marking system that solves the similar words problems in short-text answer questions as soon as the instructor evaluate.

The most of the auto-marking short answer tools are solving the similar words problems "Keyword" by the linear way, which mean that if the main words or there similarities were found in the answer then the answer is correct, without concerning that may be the similar words are not in the same level of the main words in the answer key.

The major algorithm used in this research is for using on one part of the short-text answer problem which is similar words problem, this type of problems are selected to be the domain in Fuzzy Logic design in this study.

It's for the first time **F**uzzy **L**ogic "FL" will be applied in auto-marking short-text answer, so it will be called FLASA system. FLASA will be companied with some of NLP technique as remove stop words.

There are many requirements must be fulfilling in our system are:-

- **Adequacy**: any auto-marking system must be accurate as instructor evaluation.

- **Flexibility**: any auto-marking system should be very flexible; this means that the instructor can control many parameters in evaluation such as defining the rules results and weights of synonyms words etc…...

- **Scalability**; scalability means basically how many main words in the key answer that can be evaluated by the auto-marking system. Our FLASA algorithm was tested for up to 5 main words, and it will be tested later for more n words, which will be as a future works.

- **Performance**; Performance is a necessary condition in every type of computing environment. It has special consideration in FLASA auto-marking system environments. the main consideration is the time to fill all FLASA rules; this will be done for one for each question. The time consumer depends on the number of the rules, this because the instructor's need to fill all rules results manually, all rules are defined automatically by the system and the results of these rules are defined by the instructor. The number of rules is 3^n , where n is the number of main words in the key answer.

3.2.2   Requirements and Analysis Phase.

Requirements and analysis phase is one of the important phase of FLASA design, in which the domain requirement and all methods of similar words problems system design will be clearly determined.

In this phase we must study existing auto-marking techniques, and try to reuse most of available techniques knowledge such as statistical techniques and simple keywords analysis. The requirements phase will capture all valid requirements and outline an ideal system of auto-marking system. This phase consists of two main activities: The capture requirements activity and the analysis activity. The requirement model will specify the requirements of designing auto-marking system and the analysis model will outline the main concepts of the system.

3.2.3 FLASA Design and Structure.

The FLASA design phase is sub-process of the fuzzy logic development process consists of architectural design and detailed design.

In design process the generic design solutions should be identified and that by:

- Approve the design solutions by prototyping approach.

- During the architectural design phase, we will identify the objects needed to implement the system, and the way the objects collaborate.

- Dividing the system into subsystems during this phase.

In next chapter we will introduce FLASA structure or design in details.

3.2.4 FLASA Implementation Phase.

The implementation phase follows the design phase, where all attributes and methods are identified and described. Simple implementations using simulations have been included in our work, as will be shown in chapter 4.

3.2.5   FLASA Testing Phase.

The last phase is the unit testing phase. When performing a unit test, only one unit is tested at a time. The idea is that a well defined unit with well defined responsibilities is tested so it is verified that the unit will fulfill the requirements imposed on the unit. In FLASA we test our design through building a simulator in order to validate the accurate of this design.

CHAPTER 4

FLASA DESIGN AND PROTOTYPE IMPLEMENTATION

4.1   FLASA Design.

This section describes the general structure of FLASA model, the FLASA is composed of two stages, normal method and fuzzy logic method, as shown in figure 4.1. The first stage is called normal way, which is an extension of traditional auto-marking system which solves similarity words problems as in Auto-marking algorithm that was designed by UCLES-Oxford University research [19]. It is described as; choose a set of keywords from the answer key, i.e. all essential/salient keywords that occur in the marking scheme, and a set of synonyms or similar words for each keyword. Some words weigh more.
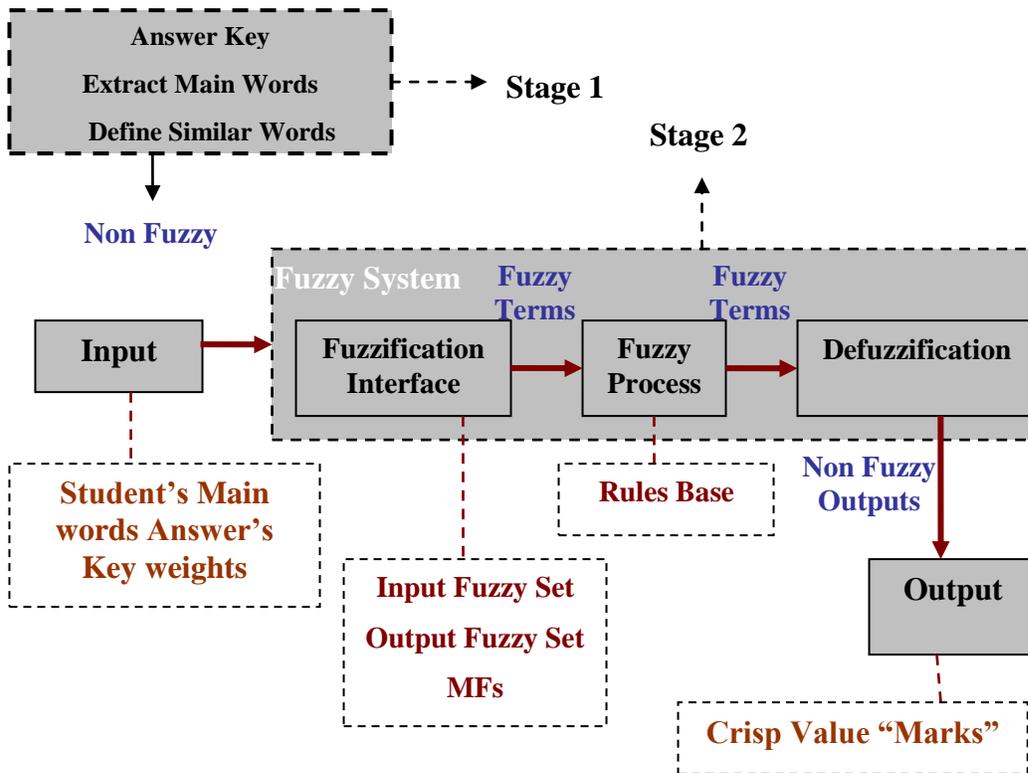
Figure 4.1: FLASA Structure .

Other systems that evaluate word use across text and documents use techniques such as **S**ingular **V**alue **D**ecomposition (**SVD**) to evaluate synonym relationships between words across texts. These methods have also been applied to essay scoring [21].

Second stage is for using FL algorithm in processing and analysis each words or similarity weights and then having the crisp number which is the answer marks. These two parts will be studied as follow.

4.1.1   Normal "Keyword" Method.

This part of FLASA presents how to extract the main words and their similarities from the key answer or references texts which were described by the instructors. This method is shown in figure 4.2, and it's described as follow; all of these steps can be done automatically as shown in next flowcharts.

- Extract all main words from key answer of the question. Here the system will separate all words in the key answers and the instructor just chooses the main words.

- The system will define all similar words in each question of the main words in the key answer, by either using locally developed DB as shown in figure B.8 "FLASA WORDS SIMILARITY" file, or defined by the instructors.

- The instructor will be defining the weight for each main word and their similarities.

- The system will extract all main words in the student's answers, by comparing it with main words in key answer defined in step one in this stage.

- The system will define the final mark by making linear summation for the weights of words in the student's answers as in formula 4.1

So if key answer with Word$_1$, Word$_2$…. and Word $_n$, and the student's answer having Word$_1$ or similar$_1$, Word $_2$ or similar $_2$…. Word $_n$ or similar $_n$, then the corresponding number for this answer is calculated as:

$$W= \sum_{i=1}^{n} M_i$$

(4.1)

Where M_i is the mark of the word$_i$ or similar$_i$. n is number of key words, and W is the corresponding number for the answer string.
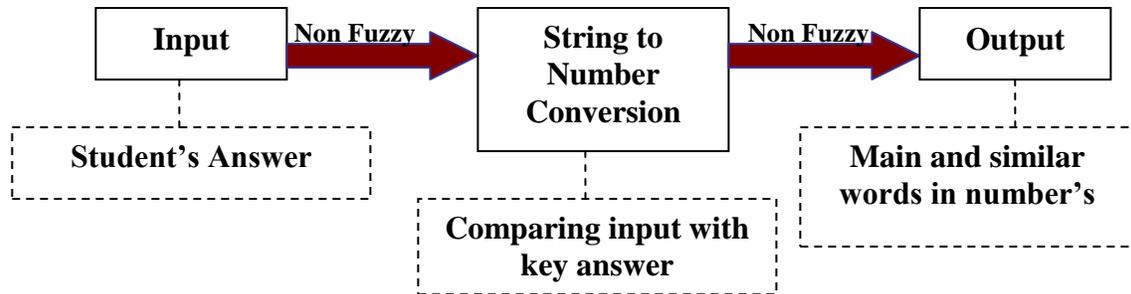
Figure 4.2: Design of Stage1.

As discussed before the final mark of the students answer, by using just Normal method which described above can be obtained, but there is need for making comparison between keyword method, and normal method and normal method using FL algorithm.

4.1.2 Fuzzy Logic "FL" Method

The steps of FLASA are shown in figure 4.1.it is will known that the most important step in a system or algorithm is to define inputs and the outputs of the system, here the main word's key answer weights or similar weights of the student is obtained from the first part of

FLASA. But there is a need to implement fuzzy system instead of summarized the weights to have a crisp or mark number for the student answer as the output of the system. The following sections will describe our algorithm carefully as previously shown in the figure 3.1.

The number of inputs depends on the number of the main key words for the question which defined by the instructor. So if we have the answer key contain three main words then we have three inputs for our system. And the number of output is fixed for any number of inputs, which is the answer mark.

4.1.2.1 Fuzzification.

Fuzzification step is the most important step in any fuzzy algorithms as in our FLASA algorithm, which is converts input number into linguist words, as it discussed in chapter 2. To be fuzzy input to the next step of our algorithm, this step called Fuzzification of the input. This step will be divided into 2 parts.

1.   *Define input fuzzy set and output fuzzy set.*

2.   *Define the membership functions for input and output fuzzy set.*

In first part we must be careful in choosing linguist words to be compatible with our system for the input and the output fuzzy set, here we have chosen three words to be a member in our input fuzzy set which is: ***Low***, ***Middle***, and ***High***, Low indicate to the small mark, Middle indicate to the medium mark and High indicate to the high and full mark for each one of the words in the key answer.

Five words to be members in our output fuzzy set which is: *Zero* to indicate to zero mark, *Low*, *Middle*, *High* and *Full* to indicate to the full mark. Each one of these words has a membership function to describe behavior.

We need to describe a membership function for each one of the input fuzzy set members, to describe the members behavior, this because each input may be need to be replaced by one or two words. But which words the membership function can be described and which words can be replaced with confidence of each linguistic words. Figure 4.3 illustrate the parts "scenario" of this step.
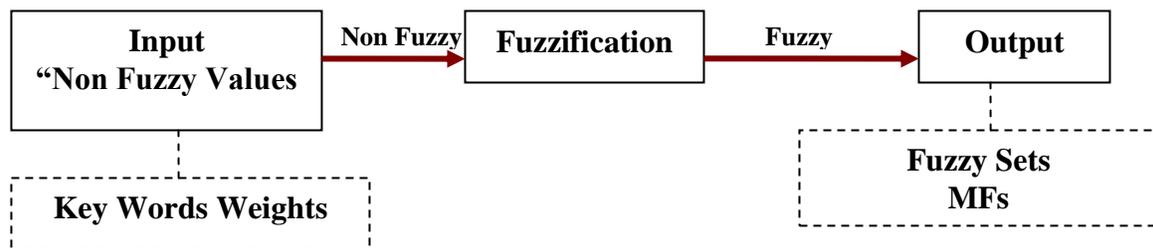
Figure 4.3: Fuzzification Design.

After we define the fuzzy set "linguistic value" for our input and output, the next part of this step is to define the membership function for each member in the input and output fuzzy set. This membership function is the method to convert our input number which is non fuzzy input into linguist input fuzzy value from fuzzy set {Low, Middle, High} for input and {Zero, Low, Middle, High, Full} for the output.

It is crucial to define in some way how to go back and forth between the description of input in numbers and the description of input in words. This stage is the most important step in our algorithm and our system. In this stage we will define the strategies of the algorithm to solve our problem, like defining the simple function and defragmenter the main interval which will be defined in this part into sub small interval each one are membership function as in the *Figure 4.4*.

Figure 4.4 illustrates the features of the *triangular* membership function for input fuzzy set, which is used in this example because of its mathematical simplicity. Other shapes as trapezoidal can be used but the *triangular* shape lends itself to this illustration and very clear to describe our algorithm in graphics way.

In the FLASA usually will use the term *confidence* rather than grade of membership as described in last chapter. This permits us to use ordinary language in describing things in a precise way and we measure the confidence "grade" that the member belongs to the fuzzy set as a number ranging from zero (absolutely false) to 1 (absolutely true).

We can calculate the confidence of each one of the members of input fuzzy set by the following functions as described in figure 4.4.

As we see in the *figure 4.4*, we have Y axis which denote to the confidence of our members which is ranging from zero (absolutely false) to 1 (absolutely true). The X axis is the weights of each word in the answer, which is ranging from 0 to 3. This rang is fixed for our algorithm, and does not depend on any parameter of our algorithm.

After we describe the membership functions for the input fuzzy set, we have to describe the membership functions for the output fuzzy set. The following figure describes the classification of the output fuzzy set which is {Zero, Low, Middle, High, Full}. Here in the output membership function figure we have the y axis which describe the confidence of the out member, and x axis which is from 0 to N, where N is the full mark of the question and it is calculated as number of main words "z" products by 3, where 3 is the maximum weight of the main key words from the key answer.

For example if we have 3 main words as the key answer then the x axis N=3 * 3 = 9, if z= 4 then N=4 * 3 =12 and so on. So in this case the output value depends on the number of input which is the key words.

$$
\text{Low=} \begin{cases} 1 & \text{IFF} \quad \text{W} <=0 \\ (1.5 - W) / 1.5 & \text{IFF} \quad 0 < W < 1.5 \\ 0 & \text{IFF} \quad W >= 1.5 \end{cases}
$$

$$
\text{Middle =} \begin{cases} 0 & \text{IFF} \quad 2.5 >= W <= 0.5 \\ (W - 0.5) & \text{IFF} \quad 0.5 < W < 1.5 \\ 1 & \text{IFF} \quad W = 1.5 \\ (2.5 - W) & \text{IFF} \quad 1.5 < W < 2.5 \end{cases}
$$

$$
\text{High=} \begin{cases} 0 & \text{IFF} \quad W <= 1.5 \\ (W - 1.5) / 1.5 & \text{IFF} \quad 1.5 < W < 3 \\ 1 & \text{IFF} \quad W >= 3 \end{cases}
$$

Figure 4.4: Possible Fuzzy Quantization of the range [0, 3] by triangular shaped.

Figure 4.5: Possible Fuzzy Quantization of the range [0, 9] by triangular shaped.

$$
\text{Very Low} = \begin{cases} 0 & \textbf{IFF} \quad \textbf{X >= N/3} \\[2em] \textbf{(1 – X/3)} & \textbf{IFF} \quad \textbf{0 <X < N/3} \end{cases}
$$

$$
\text{Low} = \begin{cases} 0 & \textbf{IFF} \quad \textbf{N/2 <= X <= N/6} \\[1.5em] \textbf{(X – Z/2) /(Z/2)} & \textbf{IFF} \quad \textbf{N/6 < X < N/3} \\[1.5em] 1 & \textbf{IFF} \quad \textbf{X=N/3} \\[1.5em] \textbf{((3Z/2) - X) / (Z/2)} & \textbf{IFF} \quad \textbf{N/3 < X < N/2} \end{cases}
$$

$$
\text{Middle} = \begin{cases} 0 & \textbf{IFF} \quad \textbf{N/1.5 <= X <= N/3} \\[1.5em] \textbf{(X – Z) /(Z/2)} & \textbf{IFF} \quad \textbf{N/3 < X < N/2} \\[1.5em] 1 & \textbf{IFF} \quad \textbf{X=N/2} \\[1.5em] \textbf{(2Z - X) / (Z/2)} & \textbf{IFF} \quad \textbf{N/2 < X < N/1.5} \end{cases}
$$

$$
\text{High} = \begin{cases} 0 & \textbf{IFF} \quad \textbf{N/1.2 <= X <= N/2} \\[1.5em] \textbf{(X – 3Z/2) /(Z/2)} & \textbf{IFF} \quad \textbf{N/2 < X < N/1.5} \\[1.5em] 1 & \textbf{IFF} \quad \textbf{X=N/1.5} \\[1.5em] \textbf{(5Z/2 - X) / (Z/2)} & \textbf{IFF} \quad \textbf{N/1.5 < X < N/1.2} \end{cases}
$$

$$
\text{Very High} = \begin{cases} 0 & \textbf{IFF} \quad \textbf{X<= N/1.5} \\[2em] \textbf{(X – 2Z)/ Z} & \textbf{IFF} \quad \textbf{N/1.5 <X < N} \end{cases}
$$

Figure 4.6: the membership functions for the output fuzzy set.

4.1.2.2 Define Rules Base.

In the last sections we described our input and output fuzzy set for our system, in this section we have to define our rules for our algorithm, these rules are mapping between inputs fuzzy set and outputs fuzzy set, these rules are called rules base. These rules mapping the input fuzzy set with the output fuzzy set.

These rules are the most important term for training our system; it will be created automatically from the system. The instructor can put the results of each rule to have the final results from the rules.

Here in our algorithm, as we know from chapter 2, the number of rules depends on number of inputs and number of members in the input fuzzy set. The number of rules is calculated by the following function:

Number of rules $= Z^n$.   (4.2)

Where Z is the number of members in the input fuzzy set which is 3 members in our algorithm, and n is the number of inputs which is the number of main words in the key answer.

In our algorithm the rules format looks like:-

If I_1 is f_{c1} and I_2 is f_{c2} and …. In is f_{cn} then Mark is $O_{min(c1,c2…,cn)}$. (4.3)

Where I is the input value, n is the number of inputs, f one of input fuzzy set members {Low, Middle, High}, c is the confidence of each one of the members of the fuzzy set," each one of

the members of fuzzy set has its own membership function as in *figure 4.4* from these membership functions we can calculate the confidences of each one of the members for each rule defined in this part", and O is one of the output fuzzy set members {Zero, Low, Middle, High, Full}. For logical "and" operations using fuzzy sets the resulting membership functions are defined as the minimum of the values of the memberships on the component sets as proposed by Lotfi Zadeh [24] .

Summary:-

- Defuzzification of the conditions of each rule and assigning the outcome of each rule the minimum Membership Value "MV" of its conditions multiplied by the rule weight.

- Assigning each outcome the maximum MV from its fired rules.

- Fuzzy inference will result in confidence factors (MVs) assigned to each outcome in the rule base.

4.1.2.3 Defuzzification.

In the last sections, we left off with the inference engine producing fuzzy output response magnitudes for each of the effective rules. It must be processed and combined in some manner to produce a single crisp (defuzzify) output.

Defuzzification is the reverse process of fuzzification. We have confidences in a fuzzy set of word descriptors as we define in our rules base, and we wish to convert these into a real number. This may be necessary if we wish to output a number to the user. In our system for example, we will probably want to tell the instructor's how many grades of the student's answers are. To determine exactly what is the answer marks, we first combine the results

from all rules into the outcome fuzzy set and then transform this composite fuzzy set to a crisp number by using one of the defuzzify methods.

There are many methods to defuzzify the rules, as discussed in chapter 2, we will use the centroid method "finds the point at which the membership function is at center of the graph" in our algorithm.

- **Centriod Method.**

The defuzzification of the data into a crisp output is accomplished by combining the results of the inference process and then computing the "fuzzy centroid" of the area as in *figures 4.7* and *4.8*. The marked strengths of each output member function are multiplied by their respective output membership function center points "Confidences" and summation. Finally, this area is divided by the sum of the marked member function strengths "Confidences" and the result is taken as the crisp output.

OUTPUT = sum (representative value "corresponding" * confidence) / sum (confidences)
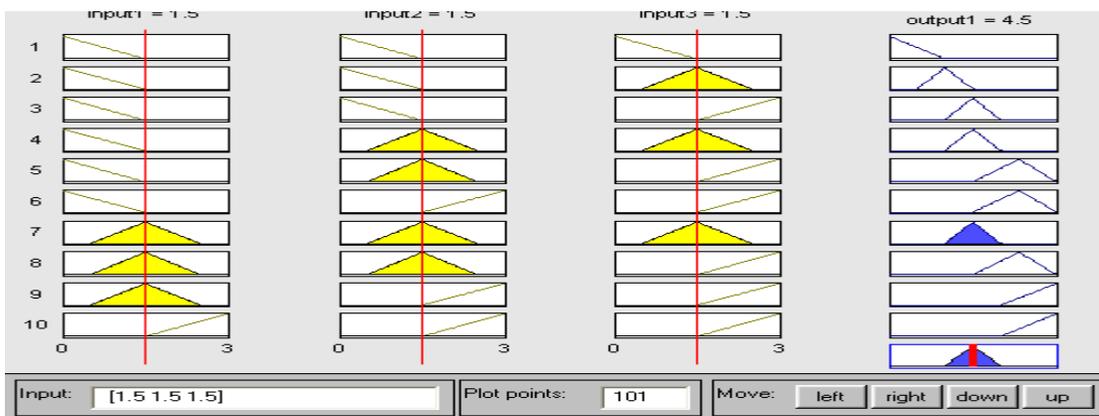
Figure 4.7: Rule extraction for the deffuzzification methods.
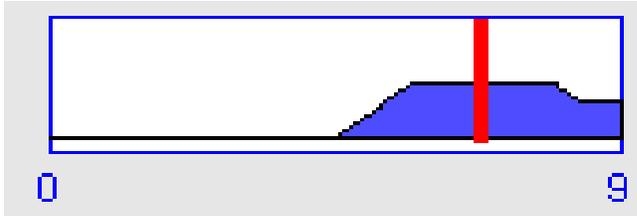
Figure 4.8: Centroid method for the output fuzzy set.

As mentioned early there are many methods to calculate the crisp number as output value "defuzzification" , as max centroid method, is that same of the centroid method except that we take the max of the scaling line instead of taking the center of scaling graph.

4.2 FLASA Procedure and Architecture.

In this section we will introduce our algorithm by procedural way. We can summarize the FLASA algorithm as the following steps, as we know from last section, we divide our algorithm into two parts, one for the normal method and the second for the FL method, and each part has its own steps, as following procedure:

- **Stage one "Normal method".**

1. *Extract all main words from the key answers "references text" of the questions and put them in set called S_{main}.*

S_{main}= {W1, W_2,…,W_n} where n is the number of main words in the answer key, which is the number of input in Fuzzy Logic methods, as we describe in last sections of this chapter.

2. *Define all synonyms or similar words for each word and their weight and put the results in a new set called S_{syn}.*

Notes" the system will define most of similar words for each word in S_{main}, by using any

synonyms methods, like Microsoft Words Synonyms.

The flowchart of first two steps of this stage is shown in figure 4.9.

3. Extract all words from the student answers "candidate text" that founds in S_{syn}. And put

it in set called S_{answer} . after this step we have an input for the second part of FLASA. The

flowchart of this step is shown in figure 4.10.

- **Stage 2 FL method**.

a fuzzy logic stage is a **M**ultiple **I**nput **S**ingle **O**utput **S**ystem (**MISO S**ystem), Multiple

means that, 3 Inputs for 3 main words, 4 Inputs for 4 main words, 5 Inputs for 5 main words,

etc. Single output which is the words' mark.

1. Define the input and output fuzzy set.

2. Define the input output membership function. Here we depend on the number of main

words found in S_{main} which is n.

The flowchart of first two steps of this stage is shown in figure 4.11.

3. Define the rules base for our input which is Z^n where Z is the number of member of input

fuzzy set which is a constant number equal 3, and n is number of words in S_{main} .

The flowchart of this step is shown in figure 4.12.

4. Define the confidences for each words in S_{syn} by applying membership function found in

step 2 , and put the results in new set called S_{Conf} .

the flowchart of this step is shown in figure 4.13.

93

5. *Define all rules for all elements in S_{conf}.*

6. *Combine the results from all rules into the outcome fuzzy set and then transform this composite fuzzy set to a crisp number by using one of the defuzzification methods.*
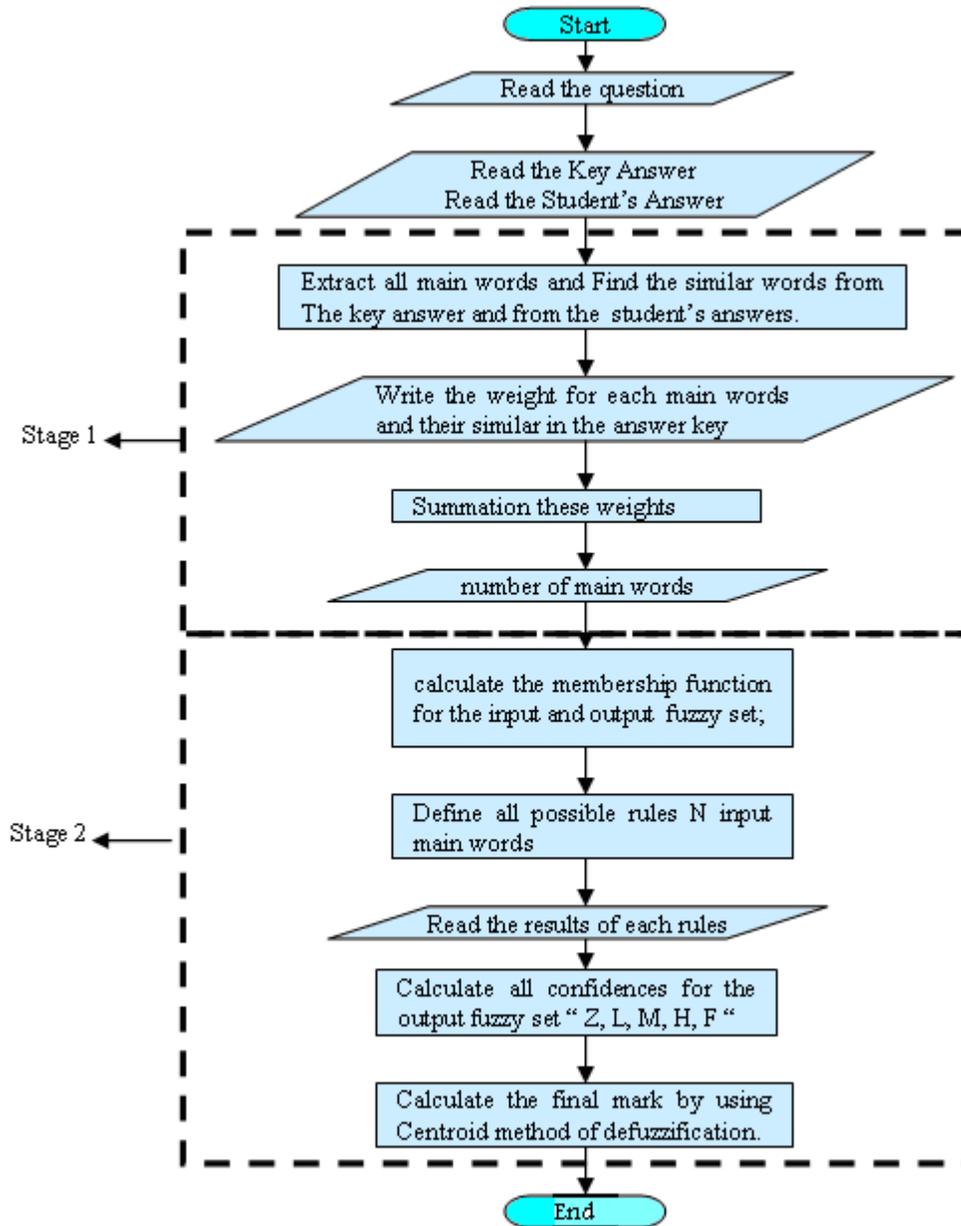
Figure 4.9: System Flowchart.
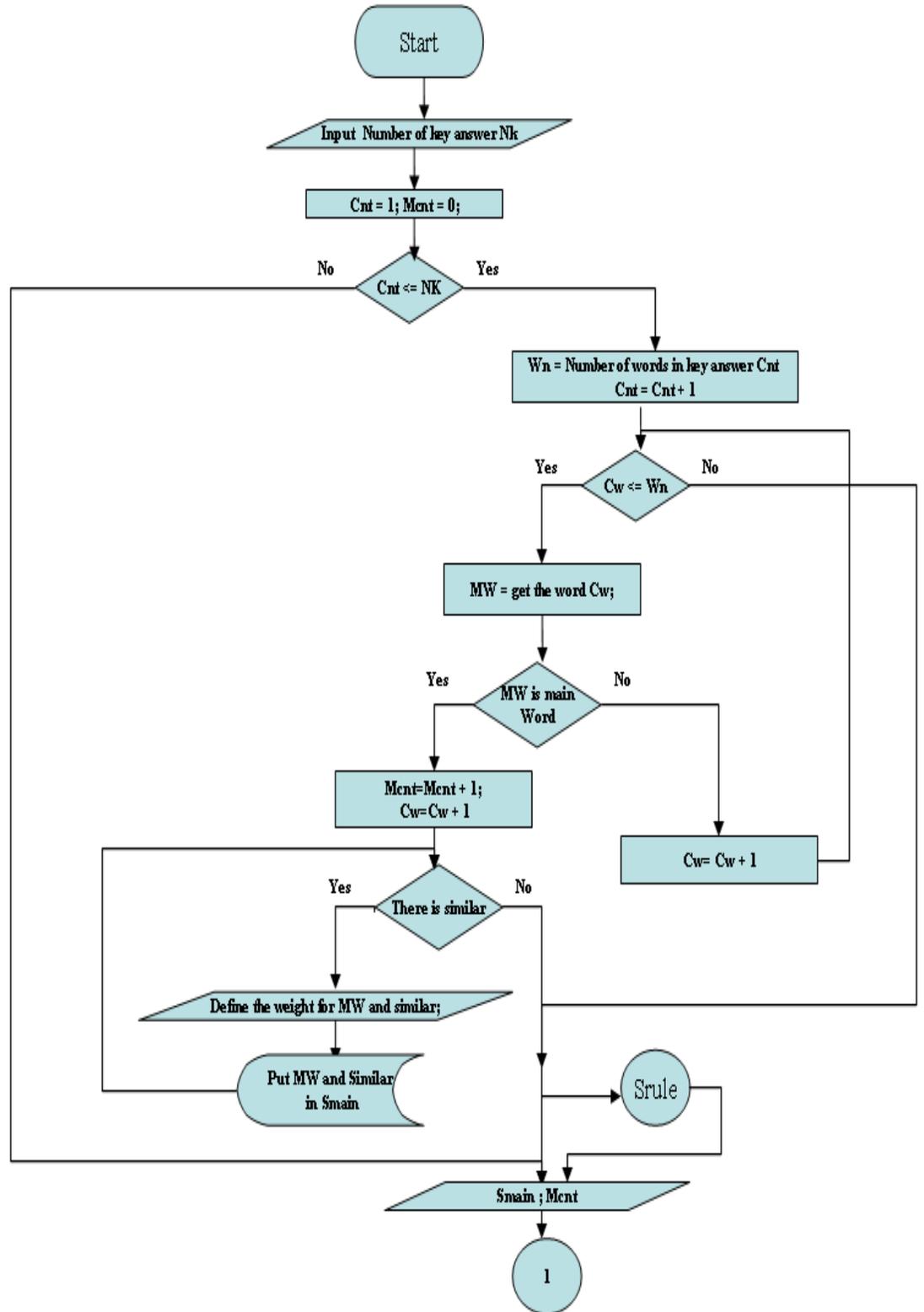
Figure 4.10: *Flowchart for the first 2 steps of FLASA stage 1.*

95
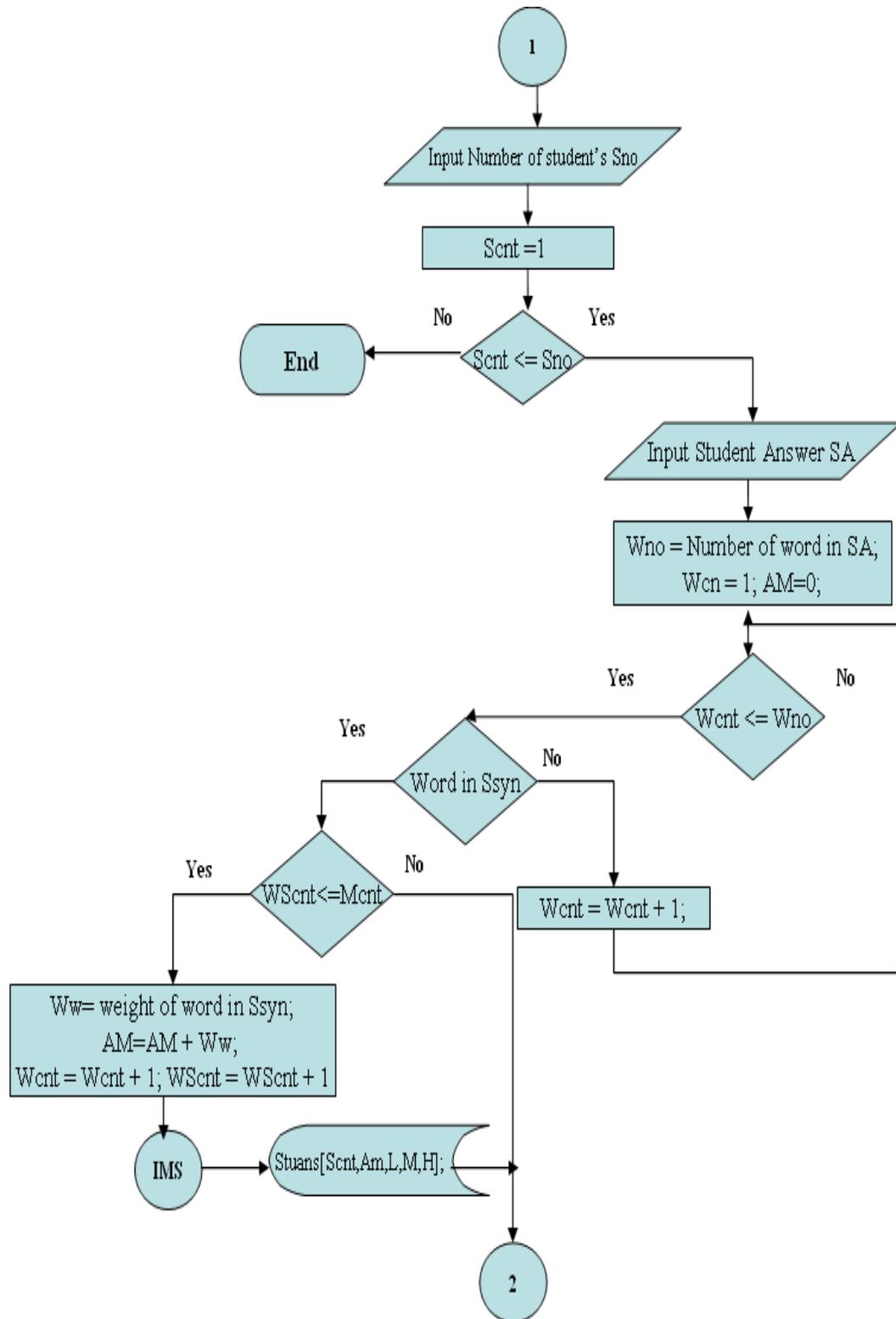
Figure 4.11: Flowchart for extraction of words in the student answers.
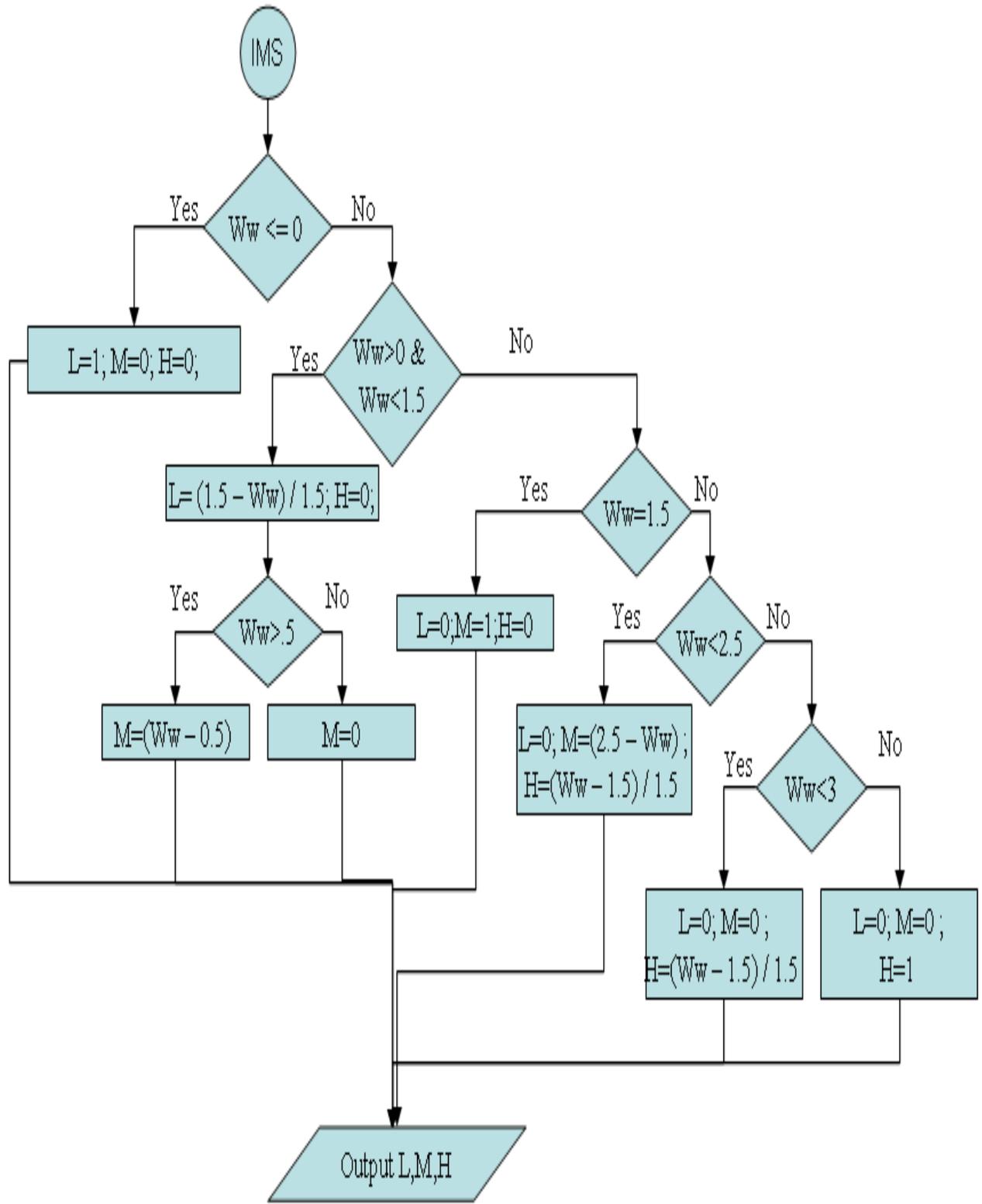
Figure 4.12: Flowchart of define the input fuzzy membership function.
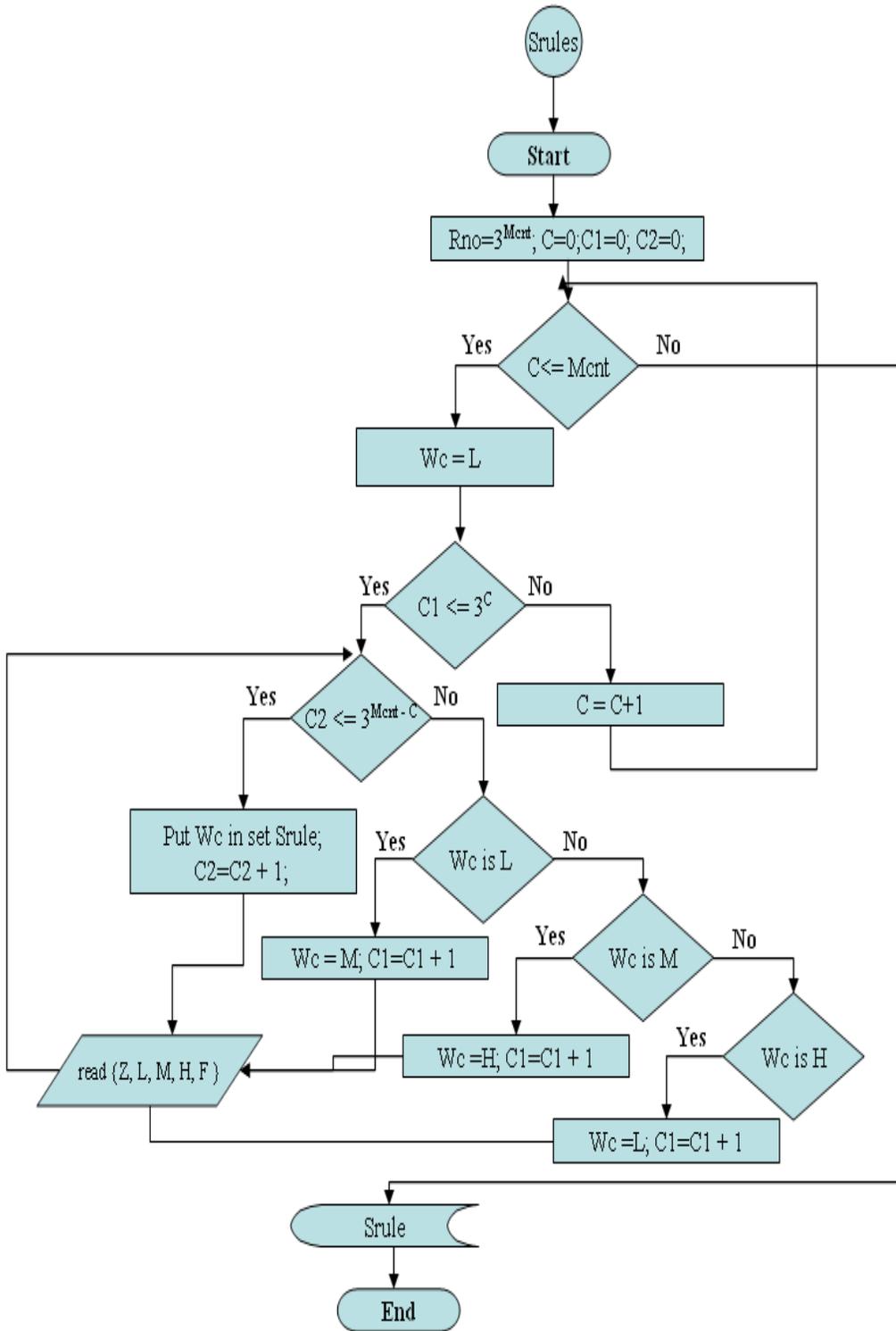
Figure 4.13: Flowchart for the Rule base step in FLASA.
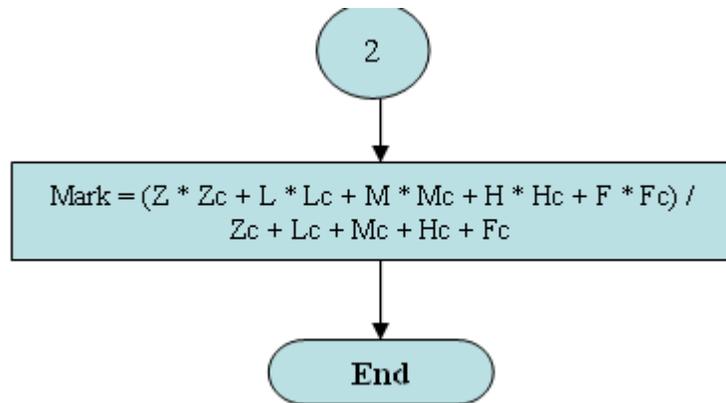
Figure 4.14: Flowchart for Centroid Defuzzification Method.

4.3 FLASA Prototype.

We have accomplished our simulation by using Matlab's R12 software to construct the rules base in stage 2 of FLASA. And using **Oracle8i** and **Developer6i**, to design a prototype program for our algorithm. Using Oracle8i as a database engine to store the huge number of words and there similarity words, as a dictionary, this because the capacity of storing data in oracle8i is unlimited, which is unlike other system like SQL server, and access …etc, which have limited capacity. While Developer6i is used to make an interface for our algorithm, this is due to the compatibility between oracle8i and developer6i. First example containing a careful detailed discussion for the developed algorithm.

1)   The instructor writes the questions of special course in the questions place, and their key answers in the key answer place, as shown in figure 4.15. It is important to note that the RDB in this case is *a set of related data*.

99

2)   The system will remove all stop words and special characters in the key answer, and writes all main words in the Main Words place in left bottom corner of the figure 4.15. The stop words and special characters will be removed from the key answer, using some of NLP technique.

3)   After collecting all the main words of the key answer, the system will display all similar words for main words, as in right corner of figure 4.15, and then instructor will put the weight for each similar word in the screen.

4)   After completion the first 3 steps it's the time to deal with student's answers; the student's answer can be written as in figure 4.16. Thus a special screen can be established through the internet to let the students write their answers.

5)   After the answers were written, the system will do the same thing as in key answer, remove all stop words and write all main words in the answer in the main words side, and then the system will put the weight for each word.

6)   Now all main words weights can be summarized and having the final marks. This way is called normal method.

7)   The system will display all possible rules for the fuzzy logic method; this step is shown in figure 4.17. The instructor just define the results of each condition I the fuzzy rules.
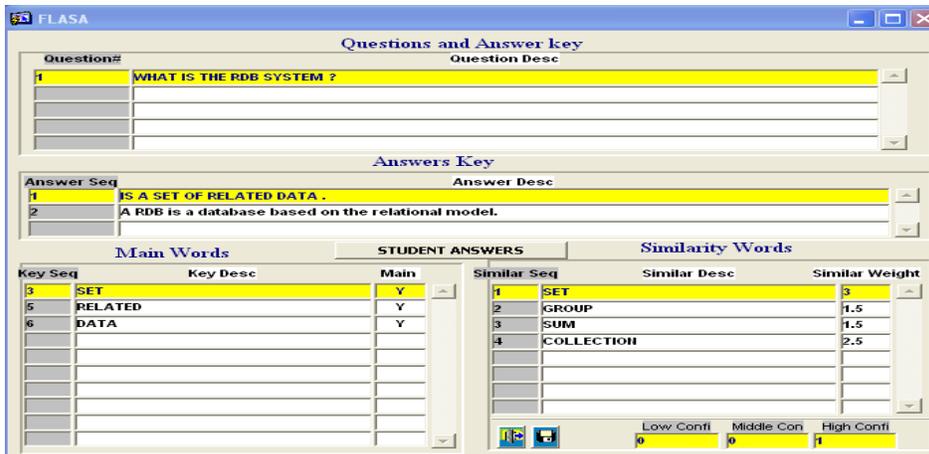
Figure 4.15: FLASA screen for questions and answers key and their similarity words.
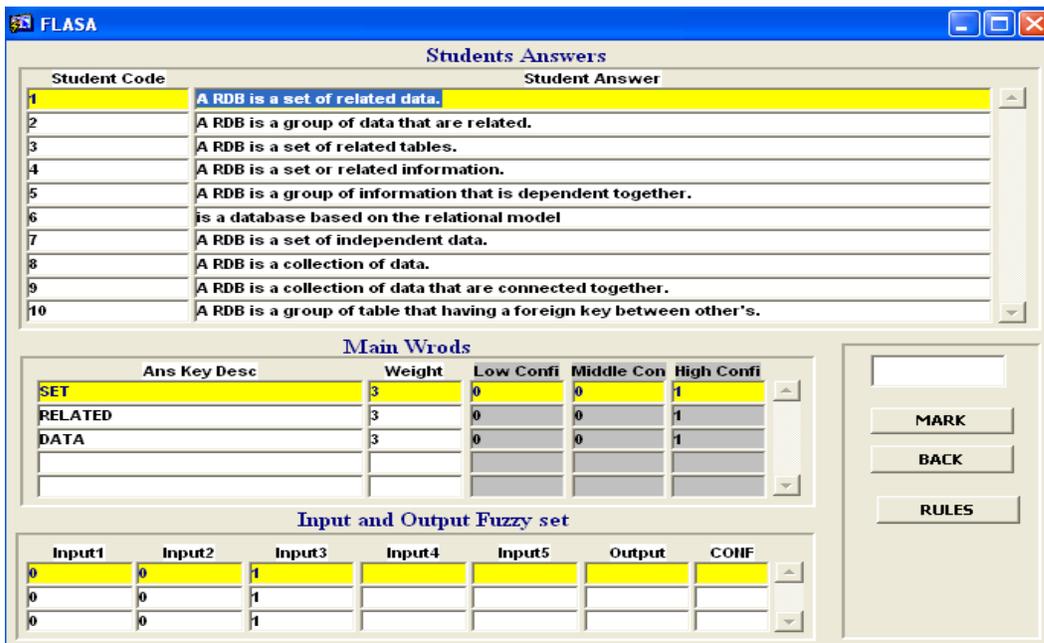
Figure 4.16: Student's answers and input output fuzzy set.

Number of Rules

Key Number	Key Desc	Rule Number
3	THREE MAIN KEY WORDS FOR THE ANSWER.	27

Rule Seq	Input1	Input2	Input3	Output Seq	Rule Desc
1	LOW	LOW	LOW	ZERO	If W1 is L and W2 is L and W3 is L then the Mark is Zero
2	LOW	LOW	MIDDLE	LOW	If W1 is L and W2 is L and W3 is M then the Mark is Low
3	LOW	LOW	HIGH	MIDDLE	If W1 is L and W2 is L and W3 is H then the Mark is Medium
4	LOW	MIDDLE	MIDDLE	MIDDLE	If W1 is L and W2 is M and W3 is M then the Mark is Medium
5	LOW	MIDDLE	HIGH	HIGH	If W1 is L and W2 is M and W3 is H then the Mark is High
6	LOW	HIGH	HIGH	HIGH	If W1 is L and W2 is H and W3 is H then the Mark is High
7	MIDDLE	MIDDLE	MIDDLE	HIGH	If W1 is M and W2 is M and W3 is M then the Mark is High
8	MIDDLE	MIDDLE	HIGH	HIGH	If W1 is M and W2 is M and W3 is H then the Mark is High
9	MIDDLE	HIGH	HIGH	FULL	If W1 is M and W2 is H and W3 is H then the Mark is Full
10	HIGH	HIGH	HIGH	FULL	If W1 is H and W2 is H and W3 is H then the Mark is Full

Figure 4.17: Rule Base in Fuzzy Logic method.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Introduction

After investigated the steps of our algorithm, and how the Fuzzy Logic can be used as a new approach in this type of methods. Three examples will be provided to check this algorithm applicability.

The first example contains three key words as a key answer. The second one has four words and the third one contains 5 words. After discussing the examples the results will be discussed to come in conclusion about the reliability and credibility of this invented algorithm.

Concerning the type of questions, FLASA only work with ***closed answers questions*** which have limited number of acceptable answers, most of which will usually be anticipated by the instructor.

5.2 Keyword Method

After each experiment we compare the results of FLASA with instructor's results and Keyword method which is an automated system. In this section, the implementation of the Keyword method for evaluating student's answers is going to be depicted, and then their performance is going to be compared against the FLASA one.

According to [5], The Keyword method is based only on counting the frequencies of words occurrences. In fact, it counts the frequency of each word of the candidate text in any of the reference texts. The procedure is taken as follows:

1. *Initialize a global counter (e.g. gCounter) to 0.*

2. *Calculate the length of the candidate text and store it in lengthCandidate.*

3. *For each word in the candidate text that is found in any reference text, add one point to gCounter.*

4. *Normalize the result by dividing by the candidate text length, so that longer texts would not be better considered than shorter ones because they have less words. The formula is:*

Keyword $= gCounter / lengthCandidate$ (5.1)

This method is no longer being used as a reliable technique but it still serves as baseline for many applications, so it is suitable to be used within the invented algorithm.

5.3 Experiments.

5.3.1 **Experiment** one.

This example having a simple question from a Database Course in ALQUDS University, which has small and simple answer "3 main words" as it can be realized from the key answer. The question appears here, what is the accuracy of the algorithm under this situation?

What is RDB System?

There are two main key answers for this question as a reference text used for the keyword methods.

A RDB is a set of related data.
A RDB is a <u>database</u> based on the relational model.

The range of mark for this question is from 0 to 9 point, so we have a Full mark which is 9, and Zero mark which is 0, and Medium mark which is 4.5 as the instruction made by the instructor. Making different possible of answers from different virtual students and outlined is possible as in table (5.1).

Table 5.1 : Student's expected answers

Student#	Student Answer
Std#1	A RDB is a **set** of **related data**.
Std#2	A RDB is a **group** of **data** that are **related**.
Std#3	A RDB is a **set** of **related tables**.
Std#4	A RDB is a **set** or **related information**.
Std#5	A RDB is a **group** of **information** that is **dependent** together.
Std#6	A RDB is a **database** based on the **relational model**
Std#7	A RDB is a **set** of **independent data**.
Std#8	A RDB is a **collection** of **data**.
Std#9	A RDB is a **collection** of **data** that are **connected** together.
Std#10	A RDB is a **group** of **table** that having a **foreign key** between other's.

- Stage 1 Normal Algorithm.

- *Stage1* of FLASA is to convert all students' answers into number to be the marks of the answer. So there is a need to define a table of main words and similar words "synonyms" and

their marks. In this example there are three main words in the key answer which is *Set*, *Related* and *Data*, and each one of the main word has more than one similar word each one with its own weights, as in the following table.

Table 5.2. The corresponding value for the main words of the key answer, this can be used as reference text for the Keyword method.

Words		Mark
Set		3
Related		3
Data		3
Main Words	Similarity	Mark
Set	group	2
	Sum	1
	Collection	1.5
Related	connected	1
	Dependent	2.5
	Foreign key	1.5
Data	information	2.5
	Table	2.5
	Model	1.5

Now converting the input strings defined from in table 5.1 to number using table 5.2 and the results appear in table 5.3.

In the Normal Algorithm we have calculated the mark directly from summation of weighting of each word and similar words.

In next stage we implement our invented algorithm to have a crisp number which is the mark of the student answer, there are more than one step to have the student's marks, we will take

the main words and there similar weights as input value for the invented algorithm, and we

process these weights in some way which is close to the instructor's idea.

Table 5.3: Corresponding number of the strings of the answers.

Student#	$W_1 / 3$	$W_2 / 3$	$W_3 / 3$	Answer Mark / 9
Std#1	3	3	3	9
Std#2	2	3	3	8
Std#3	3	3	2.5	8.5
Std#4	3	3	2.5	9
Std#5	2	2.5	2.5	7
Std#6	3	3	1.5	7.5
Std#7	3	0	3	6
Std#8	1.5	3	0	4.5
Std#9	1.5	3	1	5.5
Std#10	2	2.5	1.5	6

-Stage 2 FL Algorithm.

Fuzzification:-

- ***Part1*** *of fuzzification* is to define the input fuzzy set and the membership functions of

the input system, here the input fuzzy set is {Low, Middle, High}, in the range [0, 3], which

have the membership function as shown in figure 5.1.

- ***part2*** *of fuzzification* is to define the output fuzzy set and their membership functions.

There are five members as the output fuzzy set namely; Zero, Low, Medium, High, Full, and

the membership functions as in figure 5.2. It is must to define the interval range for the

output fuzzy set.

In our algorithm as we define in last section, the output interval is from 0 to the number of key words produced to the maximum weight of the words. In this case having 3 words with weight 3 as maximum weight of key words, so the interval is [0,9], as resulted in figure 5.2.
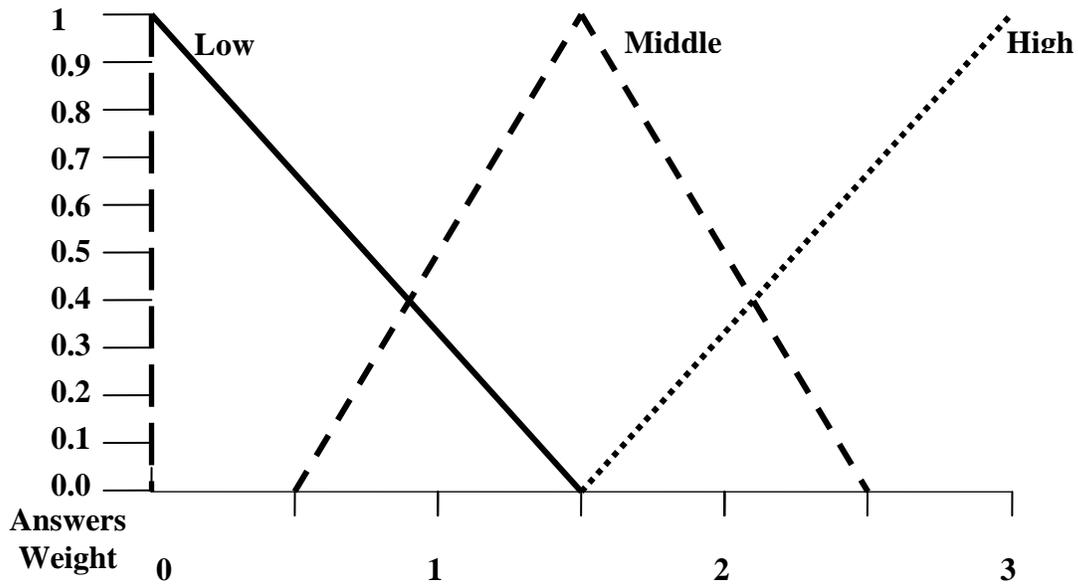
Figure 5.1: Possible Fuzzy Quantization of the range [0, 3] by triangular shaped.

$$
\text{Very Low} = \begin{cases} 0 & \text{IFF} \quad X >= 3 \\ (1 - X/3) & \text{IFF} \quad 0 < X < 3 \end{cases}
$$
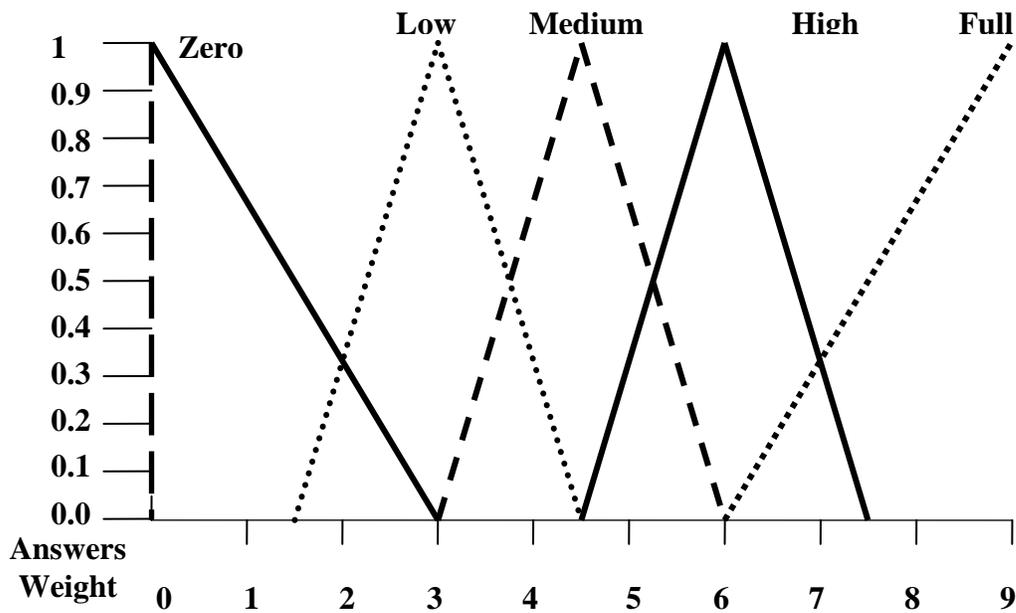
$$
\text{Low} = \begin{cases} 0 & \text{IFF} \quad 4.5 <= X <= 1.5 \\ (X - 1.5)/(1.5) & \text{IFF} \quad 1.5 < X < 3 \\ 1 & \text{IFF} \quad X = 3 \\ ((4.5) - X)/(1.5) & \text{IFF} \quad 3 < X < 4.5 \end{cases}
$$

$$
\text{Middle} = \begin{cases} 0 & \text{IFF} \quad 6 <= X <= 3 \\ (X - 3)/(1.5) & \text{IFF} \quad 3 < X < 4.5 \\ 1 & \text{IFF} \quad X = 4.5 \\ (6 - X)/(1.5) & \text{IFF} \quad 4.5 < X < 6 \end{cases}
$$

$$
\text{High} = \begin{cases} 0 & \text{IFF} \quad 7.5 <= X <= 4.5 \\ (X - 4.5)/(1.5) & \text{IFF} \quad 4.5 < X < 6 \\ 1 & \text{IFF} \quad X = 6 \\ (7.5 - X)/(1.5) & \text{IFF} \quad 6 < X < 7.5 \end{cases}
$$

$$
\text{Very High} = \begin{cases} 0 & \text{IFF} \quad X <= 6 \\ (X - 6)/3 & \text{IFF} \quad 6 < X < 9 \end{cases}
$$

Figure 5.2:- Possible Fuzzy Quantization of the range [0, 9] by trapezoidal shaped.

After obtaining the words description which is the fuzzy value. After completion the previous steps successfully the next step will investigate the rules base.

Rule Base.

Step 2 of FLASA is to define the Fuzzy Rule Base, which mapping the input fuzzy set with the output fuzzy set, as we know from the fuzzy rules base we have rule matrix with 3 dimension since we have 3 words. So our rules number is 3^3 which is 27 rules as written in *table 5.4.*

Table 5.4: All possible rules for 3 inputs main words.

AND	L & L	L & M	L & H	M & L	M & M	M & H	H & L	H & M	H & H
L	Z	L	M	L	M	H	M	H	H
M	L	M	H	M	M	H	H	H	F
H	M	H	H	H	H	F	H	F	F

As shown in table 5.4, rules with same color can be connected. This is due to; the order in our algorithm is not important, so from previous example we can reduce all possible rules from 27 rules to 10 rules, only if we have the following rules.

If W_1 is L and W_2 is L and W_3 is M then the Mark is **Low**

If W_1 is M and W_2 is L and W_3 is L then the Mark is **Low**

If W_1 is L and W_2 is M and W_3 is L then the Mark is **Low**

Is the same, because the ordered of the words in our rules is not important.

Defuzzification

The defuzzification of the data into a crisp output is accomplished by combining the results of the inference process, then computing the "**fuzzy centroid**" of the area. The weighted strengths of each output member function are multiplied by their respective output membership function center points and summed. Finally, this area is divided by the sum of the weighted member function strengths and the result is taken as the crisp output.

OUTPUT = sum (representative value "corresponding" * confidence) / sum (confidences)

After we define the Fuzzy Rule Base, we can start implements our algorithm in the example, we can describe some samples from the example to show how we can have a crisp number describes our marks of the sample.

Stud#1:-

A RDB is a **set** of **related data**.

Here we have three main words which are **Set**, **Related** and **Data**, each words has it's own weight, from the *table 5.2*, Set weight is 3, Related weight is 3 and Data weight is 3, from the input fuzzy set we see that Set is High with confidence 1, Related is High with confidence 1 and Data is High with confidence 1 so the rules which will be implement's in this case is R#10 which is:-

If W_1 is H and W_2 is H and W_3 is H then the Mark is **Full (1 & 1 & 1)**

So the result of this rule is **Full** with confidences 1, when we implements centroid methods by using matlab software, the results are shown in table 5.5 . .

Stud#2.

A RDB is a **group** of **data** that are **related**.

Here we have:-

W_1 = **Group** with weight 2, **Middle** with Confidence 0.5 and **High** with confidence 0.33.

W_2 = **Data** with weight 3, **High** with Confidence 1

W_3 = **Related** with weight 3. **High** with Confidence 1

So our rules are:-

If W_1 is M and W_2 is H and W_3 is H then the Mark is **Full (0.5 & 1 & 1) = 0.5**

If W_1 is H and W_2 is H and W_3 is H then the Mark is **Full (0.33 & 1 & 1)= 0.33**

Stud#3.

A RDB is a **set** of **related** **tables**.

W_1 = **Set** with weight 3. High with confidence 1.

W_2 = **Related** with weight 3. High with confidence 1.

W_3 = **Table** with weight 2.5. High with confidence 0.66.

So our Rules are:-

If W_1 is H and W_2 is H and W_3 is H then the Mark is **Full (1 & 1 & 0.66)= 0.66**

Stud#4.

A RDB is a **set** or **related** **information**.

W_1 = **Set** with weight 3. High with confidence 1.

W_2 = **Related** with weight 3. High with confidence 1.

W_3 = **Information** with weight 2.5. High with confidence 0.66.

So our Rules are:-

If W_1 is H and W_2 is H and W_3 is H then the Mark is **Full (1 & 1 & 0.66)= 0.66**

Stud#5.

A RDB is a **group** of **information** that is **dependent** together.

W_1 = **Group** with weight 2. **Middle** with Confidence 0.5 and **High** with confidence 0.33

W_2 = **Dependent** with weight 2.5. **High** with confidence 0.66.

W_3 = **Information** with weight 2.5. **High** with confidence 0.66.

So our Rules are:-

If W_1 is M and W_2 is H and W_3 is H then the Mark is **Full(0.5 & 0.66 & 0.66)= 0.5**

If W_1 is H and W_2 is H and W_3 is H then the Mark is **Full(0.33 & 0.66 & 0.66)= 0.33**

Stud#6.

is a **database** based on the **relational model**

W_1 = **data with** weight 3. High with confidence 1.

W_2 = **relation** with weight 3. High with confidence 1.

W$_3$ = **model** with weight 1.5. Middle with confidence 1.

The rules are:-

If **W$_1$** is M and **W$_2$** is H and **W$_3$** is H then the Mark is **Full (1 & 1 & 1)= 1**

Stud#7.

A RDB is a **set** of **independent** **data**.

W$_1$ = **data** with weight 3. **High** with confidence 1.

W$_2$ = **independent** with weight 0. **Low** with confidence 1.

W$_3$ = **set** with weight 1.5. **Middle** with confidence 1.

The rules are:-

If **W$_1$** is H and **W$_2$** is L and **W$_3$** is H then the Mark is **High (1 & 1 & 1)= 1**

Stud#8.

 A RDB is a **collection** of **data**.

W$_1$ = **Data** with weight 3. High with confidence 1.

W$_2$ = **Collection** with weight 3. High with confidence 1.

W$_3$ = **Null** with weight 0. Low with confidence 1.

The rules are:-

If **W$_1$** is H and **W$_2$** is H and **W$_3$** is L then the Mark is **High (1 & 1 & 1) = 1**

Stud#9.

 A RDB is a **collection** of **data** that are **connected** together.

W_1 = **data with** weight 3. High with confidence 1.

W_2 = **connected** with weight 1. Low with confidence 0.33 and Middle with confidence 0.5.

W_3 = **collection** with weight 1.5. Middle with confidence 1.

The rules are:-

If W_1 is L and W_2 is M and W_3 is H then the Mark is **High (0.33 & 1 & 1)= 0.33**

If W_1 is M and W_2 is M and W_3 is H then the Mark is **High (0.5 & 1 & 1)=0.5**

Stud#10.

A RDB is a **group** of **table** that having a **foreign key** between other's.

W_1 = **table** weight 2.5. High with confidence 0.66.

W_2 = **foreign key** with weight 1.5. Middle with confidence 1.

W_3 = **group** with weight 2. Middle with confidence 0.5 and High with confidence 0.33.

The rules are:-

If W_1 is M and W_2 is M and W_3 is H then the Mark is **High(0.33 & 1 & 0.66)=0.33**

If W_1 is M and W_2 is H and W_3 is H then the Mark is **Full (1 & 0.66 & 1)=0.66**

After we processing all samples in the example 1, the result will be appeared in table 5.5. in this table we have the results from the normal algorithm, and from different deffuzzification methods for the output fuzzy set, and for different membership function shape to see how the output different from one method to another deffuzzification method, and from shape to shape of the membership shape functions.

We can use table 5.2 as a references text for the *keyword* method and students answer in table

5.1 as a candidate text.

Table 5.5: Results of the normal, FLASA and Keyword Method, instructors evaluate.

| Student# | Normal Method | Triangular MF | | Trapezoidal MF | | instructors | Keyword Method |
		Centroid Method	MOM	Centroid Method	MOM		
Std#1	9	8.03	9	7.94	8.51	9	9
Std#2	8	7.86	8.28	7.94	8.51	9	6.75
Std#3	8.5	7.94	8.51	7.94	8.51	9	6.75
Std#4	8.5	7.94	8.51	7.94	8.51	9	6.75
Std#5	7	7.86	8.28	7.94	8.51	8	1.8
Std#6	7.5	8.03	9	7.94	8.51	9	9
Std#7	6	8.03	9	7.94	8.51	0	6.75
Std#8	4.5	6	6.03	6	6.03	5	6
Std#9	5.5	6	6.03	6	6.03	8	3.6
Std#10	6	6.74	6.03	6.66	6.03	8	1.8

From the results of example 1 as we see in table 5.5, we conclude that the results of our

algorithm are some how equal the results of the normal algorithm as shown in figure 5.3.

This indicates that this algorithm is highly applicable for 3 input as main words in the key

answer.

Table 5.5 show the results of evaluation answer with 3 main words as an answer key, by

using FLASA, Keyword and Normal methods and the instructor evaluate. Figure 5.3 and 5.4

summarize the experiment result.
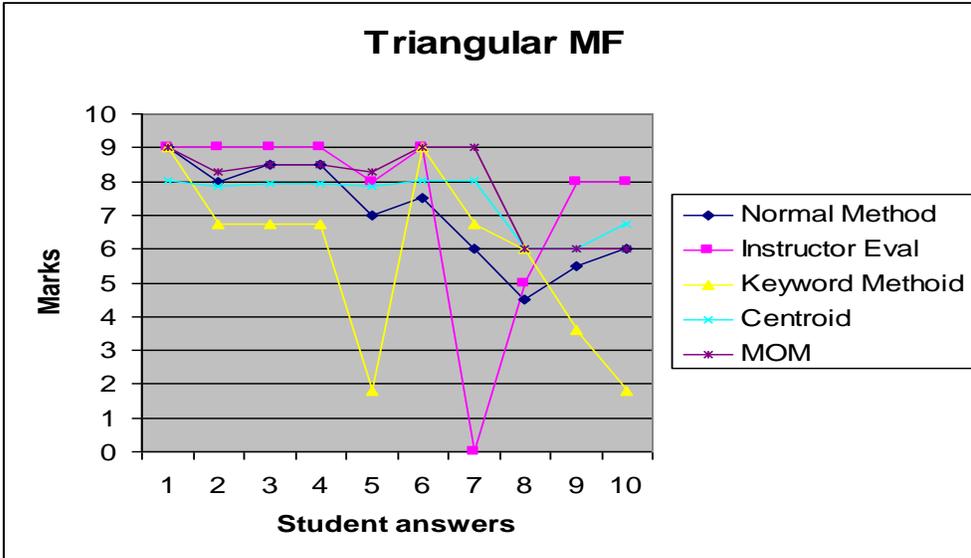
Figure 5.3 : Graph that compares the FLASA algorithm using triangular MS with the Keyword and the instructor ones, for three inputs.
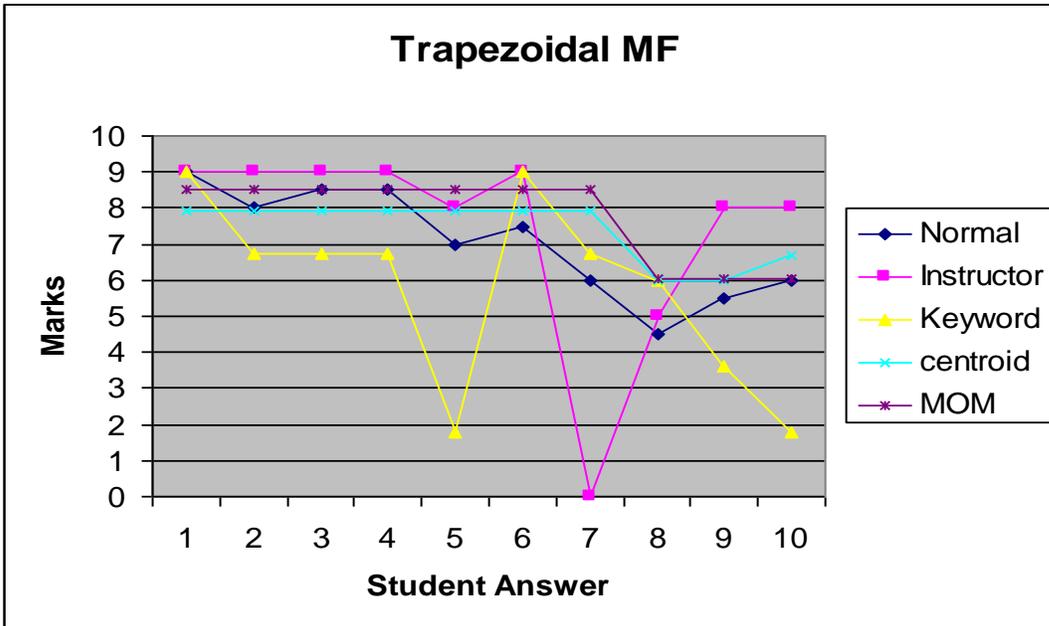
Figure 5.4: Graph that compares the FLASA algorithm using trapezoidal MS with the Keyword and the instructor ones, for three inputs.

5.3.2   Experiment 2.

In this example the key answer contains four main words. we must implement this example in Normal and FLASA algorithms. In this example we have 6 student's this is because most of student's had the same answer, but few of them got different answers.

> **Mars Polar Lander-Where Are You?**
>
> **(January 18, 2000) After more than a month of searching for a signal from NASA's Mars Polar Lander, mission controllers have lost hope of finding it. The Mars Polar Lander was on a mission to Mars to study its atmosphere and search for water, something that could help scientists determine whether life ever existed on Mars. Polar Lander was to have touched down December 3 for a 90-day mission. It was to land near Mars' South Pole. The lander was last heard from minutes before beginning its descent. The last effort to communicate with the three-legged lander ended with frustration at 8 a.m Monday.**
>
> **"We didn't see anything," said Richard Cook, the spacecraft's project manager at NASA's Jet Propulsion Laboratory. The failed mission to the Red Planet cost the American government more than $200 million dollars. Now, space agency scientists and engineers will try to find out what could have gone wrong. They do not want to make the same mistakes in the next mission. Controllers have been testing dozens of different scenarios to try and explain what might have happened to the lander. (Sources: Associated Press, CBC News Online, CBC Radio news, NASA) Copyright CBC/SRC, 1997.**

Sample reading comprehension passage with questions. News story courtesy of the Canadian Broadcasting Corporation 4 Kids site, http://cbc4kids.com/general/whats-new/daily-news.

> *What was the mission of the Mars Polar Lander?*

> **Correct sentence key:**
> Sentence 3: The Mars Polar Lander was on a mission to Mars to study its atmosphere and search for water, something that could help scientists determine whether life ever existed on Mars.
>
> **Answer key:**
> to study Mars' atmosphere and to search for water |
> to help scientists determine whether life ever existed on Mars
>
> **Sample system answer:**
> to study its atmosphere
>
> **Answer-word recall:**
> key 1: (alternative 1):   [study, atmosphere, search, water]
> system: [study, atmosphere]
> Recall: 2/5 = 40%
>
> Instructor: 2/2 = 100%

As we see there are two answers key for this question from previous example, we will take the first one, which contain 4 main words. We have to write all possibility similar words for these 4 mains words, as in table 5.6.

After we have all similar words and their weights, we can start in our example by using normal algorithm and FLASA algorithm. Our samples in this example are:

Stud#1: To study Mars' atmosphere and to search for water |

Stud#2: To learn mars characters and to search if there life or not.

Stud#3: To study mars atmosphere and seek about life.

Stud#4: To study mars atmosphere and explorer the life on it.

Stud#5: To discover mars air and rummage around the life

Stud#6: To revise mars environment and seeking if there water on mars or not.

Table 5.6: Similar words for example 2.

Words		Mark
Study		**3**
atmosphere		**3**
search		**3**
water		**3**
Main Words	**Similarity**	**Mark**
Study	**learn**	**2**
	Discover	**1**
	revise	**1.5**
Atmosphere	**Environment**	**2**
	Character	**1.5**
	Air	**2.5**
Search	**Look for**	**2.5**
	Seek	**1**
	explore	**1.5**
	Rummage around	**2**
Water	**H2O**	**3**
	Life	**1**

Normal Algorithm:-

After we implement the normal algorithm we have the following table (5.7), which contains
the results of the normal marking.

Table 5.7: the results of Stage 1, instructor, Keyword method evaluation.

Student #	W_1 / 3	W_2 / 3	W_3 / 3	W_4 / 3	Answer Mark / 12	Keyword method	Instr
Std#1	3	3	3	3	12	12	12
Std#2	2	1.5	3	2.5	9	7.2	10
Std#3	3	3	1	2.5	9.5	9.6	10
Std#4	3	3	1.5	2.5	10	9.6	11
Std#5	2	2.5	2.5	1.5	8.5	4.8	11
Std#6	1	1.5	1.5	3	7	4.8	12

FLASA Algorithm:-

Here we have four main words in the key answer, so from our algorithm we have $3^4 = 81$ rules, as we see in table (5.8) which describe the rule matrix with 4 dimensions.

The input membership functions are standard for all examples and samples as in example1, but the different in the output membership function. The membership function for the output fuzzy set in our example is the following:-

$$
\text{Very Low} = \begin{cases} 0 & \text{IFF} \quad X >= 4 \\ (1 - X/3) & \text{IFF} \quad 0 < X < 4 \end{cases}
$$

$$
\text{Low} = \begin{cases} 0 & \text{IFF} \quad 6 <= X <= 2 \\ (X - 2)/(2) & \text{IFF} \quad 2 < X < 4 \\ 1 & \text{IFF} \quad X = 4 \\ ((6) - X)/(2) & \text{IFF} \quad 4 < X < 6 \end{cases}
$$

$$
\text{Middle} = \begin{cases} 0 & \text{IFF} \quad 8 <= X <= 4 \\ (X - 4)/(2) & \text{IFF} \quad 4 < X < 6 \\ 1 & \text{IFF} \quad X = 6 \\ (8 - X)/(2) & \text{IFF} \quad 6 < X < 8 \end{cases}
$$

$$
\text{High} = \begin{cases} 0 & \text{IFF} \quad 10 <= X <= 6 \\ (X - 6)/(2) & \text{IFF} \quad 6 < X < 8 \\ 1 & \text{IFF} \quad X = 8 \\ (10 - X)/(2) & \text{IFF} \quad 8 < X < 6 \end{cases}
$$

$$\text{Very High} = \begin{cases} 0 & \textbf{IFF} \quad X <= 8 \\ (X - 8)/4 & \textbf{IFF} \quad 8 < X < 12 \end{cases}$$
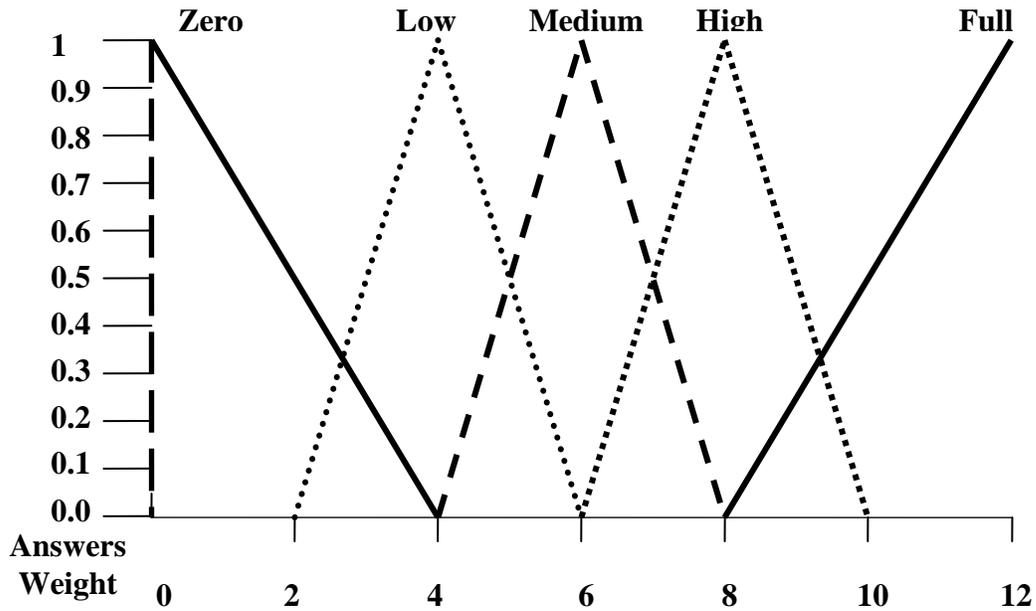
Figure 5.5:- Possible Fuzzy Quantization of the range [0, 9] by trapezoidal shaped.

As we see from table 3.8, we have all possible rules based on 4 inputs with 3 fuzzy members. As in example one we can remove all redundant rules. These rules are for any example with 4 inputs as main words.

As known from the rule base section, the results of the rules defined by the system or by the instructors. Thus writing these results by the system, due to the maximum number of the rules.

Table 5.8 The rule matrix for 4 inputs words.

&	L & L	L & M	L & H	M & L	M & M	M & H	H & L	H & M	H & H
L & L	Z	L	L	L	M	M	M	M	H
L & M	L	M	M	M	M	H	M	H	H
L & H	M	M	H	M	H	H	H	H	H
M & L	L	M	M	M	M	H	M	H	H
M & M	M	M	H	M	H	H	H	H	H
M & H	M	H	H	H	H	H	H	H	F
H & L	M	M	H	M	H	H	H	H	H
H & M	M	H	H	H	H	H	H	H	F
H & H	H	H	H	H	H	F	H	F	F

After having inputs weights for all words, and after we have a rule matrix for 4 words, and the membership functions for the output fuzzy set. We must implement one of the defuzzification methods of our algorithm to have an output marks for the students answers. The same as in fist example we will use centroid and Mea Of Max (MOM) method for defuzzification method. As illustrated in table (5.9), which have the final results of our algorithm.

Table 5.9: Results of FLASA by using different Defuzzification methods.

	Triangular MF		Normal	Trapezoidal MF			
Stude#	**Centroid**	**MOM**	**Mark/12**	**Centroid**	**MOM**	**Key method**	**Instr**
Std#1	10.7	12	12	10.6	11.3	12	12
Std#2	8.99	8.04	9	8.93	7.98	7.2	10
Std#3	9.48	11	9.5	9.69	11.3	9.6	10
Std#4	10.6	11.3	10	10.6	11.3	9.6	11
Std#5	8.99	8.04	8.5	8.93	7.98	4.8	11
Std#6	8	8.04	7	8	7.98	4.8	12
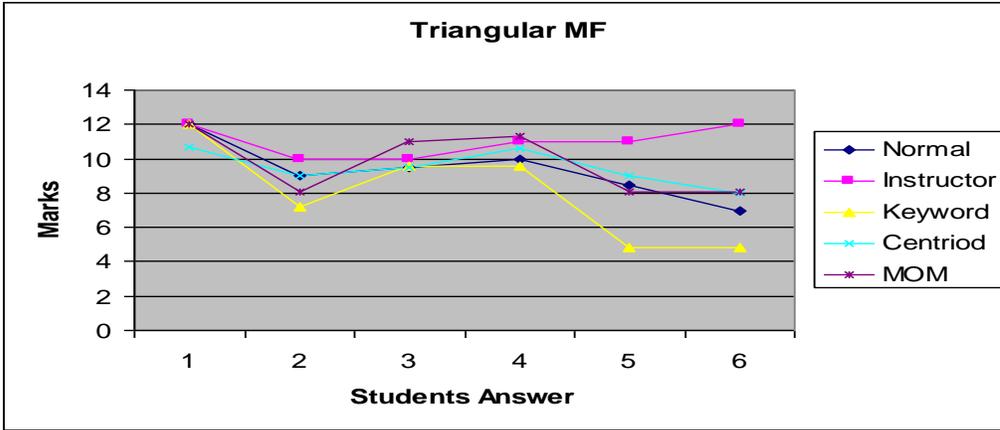
Figure 5.6: Graph that compares the FLASA algorithm using triangular MS with the Keyword and the instructor ones, for four inputs.
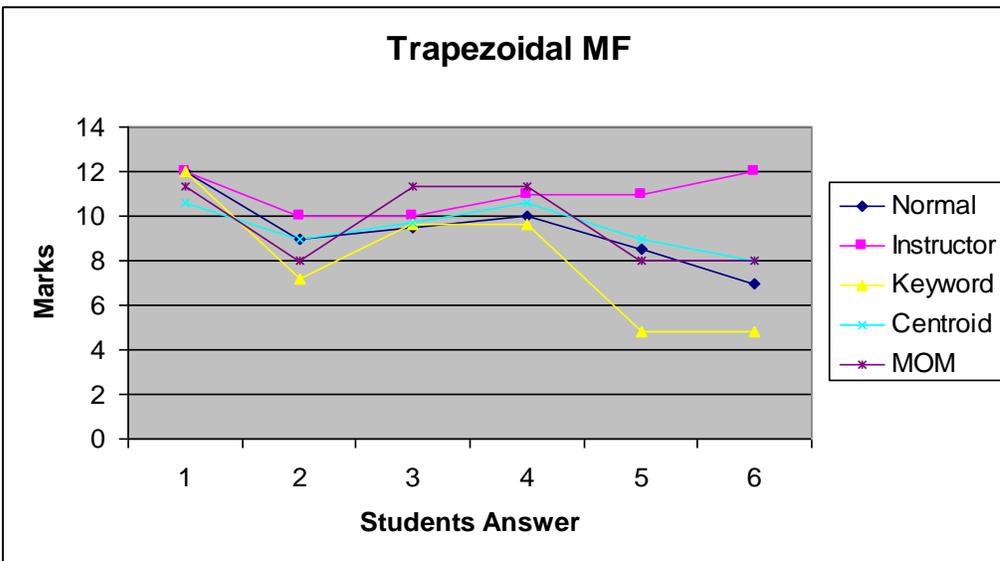
Figure 5.7: Graph that compares the FLASA algorithm using trapezoidal MF with the Keyword and the instructor ones, for four inputs.

5.3.3 Example Three.

In this example we will use a sample question with five main words in the key answer, so we have 243 rules as a rule base of our algorithm as in table 5.11. And we have an output membership functions ranged from 0 to 15 as in figure (5.8).

The question is:

What is the Bloods function?

The key answers are:-

Transport the **O2** and **food** to the **cell** and take the **CO2** from it.

The similar words table:

Table 5.10: Similar words for the example 3.

Words	Mark
Transport	3
O2	3
food	3
cell	3
CO2	3

Main Words	Similarity	Mark
Transport	**Carry**	2
	take	1
O2	**Oxygen**	2
	Air	1.5
Food	**Nutrient**	2.5
	Power	1
	Carbohydrate	1.5
Cell	**Body**	3
CO2	**Waste**	1

The student's answers are:

1. **Taking** the **food** to the **cell** and **CO2** from the cell.

2. **Carrying** the **O2** and the **food** to the **cell** and take the **waste** from it.

3. **Transport** the **food** and **CO2** to the **Cell** and taking the **waste** from it.

4. **Carry** the **oxygen** and **carbohydrate** to the **cell** and take out the **CO2** from the cell.

5. **Transport** the **Air** and the **Power** to the **body** and then taking the **waste** from it.

6. **Transporting** the **nutrient** and the **O2** to the **Cell**.

7. **Transporting** the **Food** to the **cell** and **taking** the **waste** from it.

8. **Carrying** the **oxygen** to the **cell** and **taking** out the **CO2** from it.

9. **Carrying** out the **waste** from the **cell** and **power** to it.

10. **Transporting** the **O2** and **Food** to the **cell** and **CO2** from it.

Table 5.11: Rule Base for 5 input main words.

&	L & L	L & M	L & H	M & L	M & M	M & H	H & L	H & M	H & H
L & L & L	Z	L	L	L	L	L	L	L	M
L & L & M	L	L	L	L	L	M	L	M	M
L & L & H	L	L	M	L	M	M	M	M	M
L & M & L	L	L	L	L	L	M	L	M	M
L & M & M	L	L	M	L	M	M	M	M	M
L & M & H	L	M	M	M	M	M	M	M	H
L & H & L	L	L	M	L	M	M	M	M	M
L & H & M	L	M	M	M	M	M	M	M	H
L & H & H	M	M	M	M	M	H	M	H	H
M & L & L	L	L	L	L	L	M	L	M	M
M & L & M	L	L	M	L	M	M	M	M	M
M & L & H	L	M	M	M	M	M	M	M	H
M & M & L	L	L	M	L	M	M	M	M	M
M & M & M	L	M	M	M	M	M	M	M	H

M & M & H	M	M	M	M	M	H	M	H	H
M & H & L	L	M	M	M	M	M	M	M	H
M & H & M	M	M	M	M	M	H	M	H	H
M & H & H	M	M	H	M	H	H	H	H	H
H & L & L	L	L	M	L	M	M	M	M	M
H & L & M	L	M	M	M	M	M	M	M	H
H & L & H	M	M	M	M	M	H	M	H	H
H & M & L	L	M	M	M	M	M	M	M	H
H & M & M	M	M	M	M	M	H	M	H	H
H & M & H	M	M	H	M	H	H	H	H	H
H & H & L	M	M	M	M	M	H	M	H	H
H & H & M	M	M	H	M	H	H	H	H	H
H & H & H	M	H	H	H	H	H	H	H	F
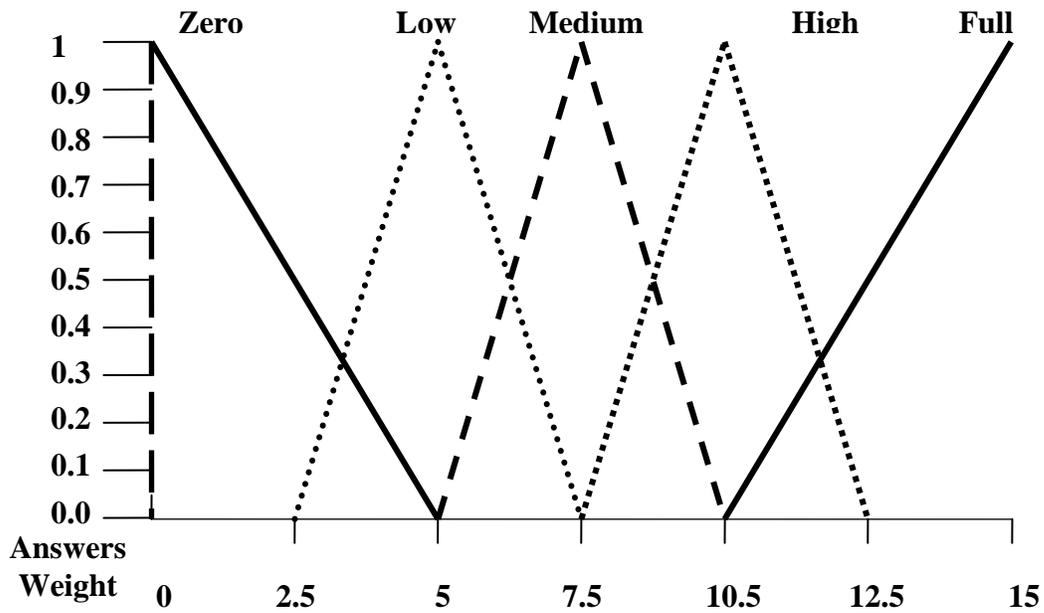
Figure 5.8 :- Possible Fuzzy Quantization of the range [0, 9] by trapezoidal shaped.

$$
\text{Very Low} = \begin{cases} 0 & \textbf{IFF} \quad X >= 5 \\[2ex] (1 - X/3) & \textbf{IFF} \quad 0 < X < 5 \end{cases}
$$

$$
\text{Low} = \begin{cases} 0 & \textbf{IFF} \quad 7.5 <= X <= 2.5 \\[2ex] (X - 2.5)/(2.5) & \textbf{IFF} \quad 2.5 < X < 5 \\[2ex] 1 & \textbf{IFF} \quad X = 5 \\[2ex] ((7.5) - X)/(2.5) & \textbf{IFF} \quad 5 < X < 7.5 \end{cases}
$$

$$
\text{Middle} = \begin{cases} 0 & \textbf{IFF} \quad 10 <= X <= 5 \\[2ex] (X - 5)/(2.5) & \textbf{IFF} \quad 5 < X < 7.5 \\[2ex] 1 & \textbf{IFF} \quad X = 7.5 \\[2ex] (10 - X)/(2.5) & \textbf{IFF} \quad 7.5 < X < 10 \end{cases}
$$

$$
\text{High} = \begin{cases} 0 & \textbf{IFF} \quad 12.5 <= X <= 7.5 \\[2ex] (X - 7.5)/(2.5) & \textbf{IFF} \quad 7.5 < X < 10 \\[2ex] 1 & \textbf{IFF} \quad X = 10 \\[2ex] (12.5 - X)/(2.5) & \textbf{IFF} \quad 10 < X < 12.5 \end{cases}
$$

$$
\text{Very High} = \begin{cases} 0 & \textbf{IFF} \quad X <= 10 \\[2ex] (X - 10)/5 & \textbf{IFF} \quad 10 < X < 15 \end{cases}
$$

After we defined the membership functions for the output fuzzy set, and defined all rules base, this making simulation by using matlab we have the following results shown in table 5.12.

Table 5.12: Results of normal and FLASA methods.

St#	Inputs					Triangular			Trapezoidal	
	$W_1/3$	$W_2/3$	$W_3/3$	$W_4/3$	$W_5/3$	Cent Method	MOM	Instrutor/ 15	nor	key
S1	3	0	3	3	3	10	10	11	12	15
S2	2	3	3	3	2	11.2	13.8	15	13	15
S3	3	0	3	3	2	11.2	10	10	11	15
S4	3	3	1.5	3	3	13.4	15	15	13.5	15
S5	3	1.5	1	2	2	9.88	10	13	9.5	15
S6	3	3	2.5	3	0	10	10	10	11.5	15
S7	3	0	3	3	2	11.2	10	10	11	15
S8	3	3	0	3	3	13.4	13.8	12	12	15
S9	3	0	1	3	2	10	10	10	9	15
S10	3	3	3	3	3	13.4	15	15	15	15
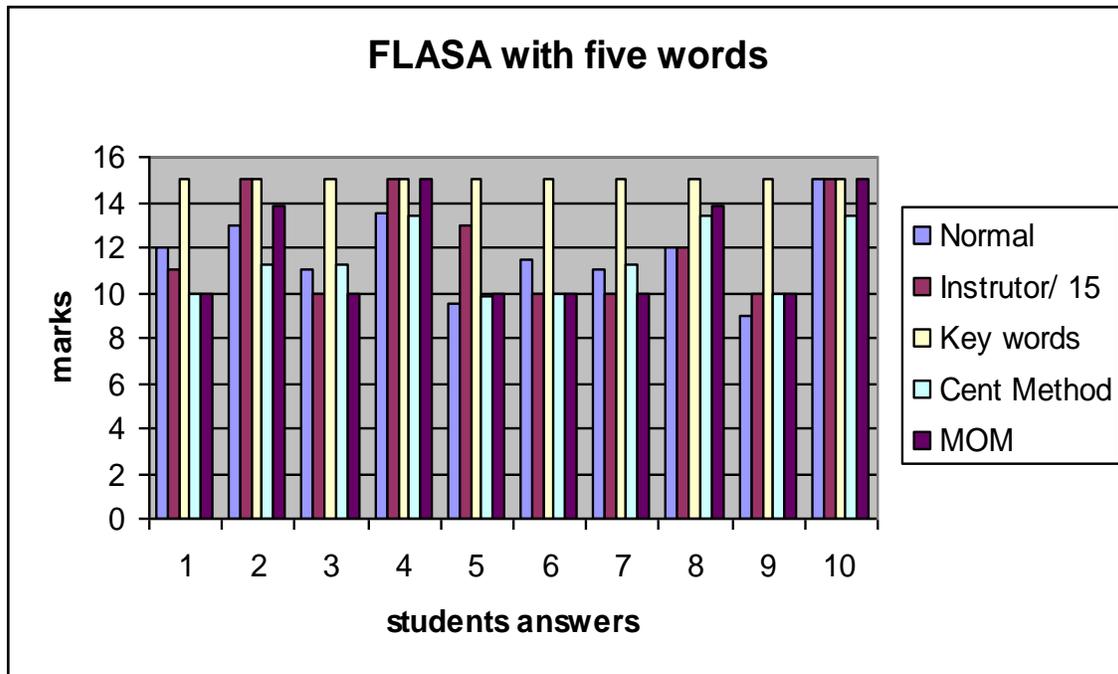
Figure 5.9: Graph that compares the FLASA algorithm using Triangular MF with the instructor ones, Norma and Keywords for five inputs

129

5.4   Discussion

From the previous results obtained. We can classify our observations into two categories: the first one is about the FLASA features as will presented in   section 5.5 and the second category about the FLASA adequacy introduce in the next subsection 5.4.1.

5.4.1   FLASA Adequacy.

It is understood that a uniform metric to give the adequacy results of CAA of free text answer systems is necessary. A reference corpus should be made available for the CAA community **[35]**. Whenever this measure and these references appear we would use it. At the same time, we have built our own corpus and analyzed FLASA adequacy by calculating the correlation between the teachers' scores and the FLASA scores as in formula 5.2. The results have been very promising as shown in Tables 5.5, 5.9 and 5.10.

$$\text{Adequacy} = \frac{\sum \text{FLASA Results}}{\sum \text{Instructor Results}} \times 100\% \qquad (5.2)$$

Table 5.13: FLASA Results.

# of main words	Centroid	MOM
3	89%	94%
4	86%	88%
5	93%	97%
Average	89	93

According to P´erez **[46]**, FLASA adequacy has also been measured by using the following procedure:

- *For each of the data sets, evaluate all the answers with FLASA.*

- *Scale the resulting scores using the linear regression equation to translate them to the teacher's scale.*

- *Calculate the deviation between FLASA scores and the teachers' scores.*

- *Generate the histogram of the deviations.*

Figures 5.3, 5.4, 5.6 and 5.7 are all an example of Histogram that shows the deviation of FLASA scores. From the figures "5.3, 5.4, 5.6, 5.7, 5.9" we can summarize the output of our FLASA algorithm by the following points:

The adequacy of FLASA is obvious from figures 5.3, 5.4, 5.6 and 5.7, that centroid & MOM methods were closer to the reference marking (instructor) than the two other systems as shown in table 5.13. However, the normal methods is pounding a mean results since it depends on the summation without distinguish between words weight distribution. While the keywords method showed the lowest efficiency between all methods due to its way of dealing with words which can be described as concerning the reference text rather than keywords weight. However, the centroid and MOM methods showed higher adequacy when keywords number increasing as shown in table 5.13.

Due to the differences that appears in figure 5.3 especially question number 7, mainly the one between the instructor mark (0 mark) and the FLASA two methods marks (8 and 9 marks) is

shown to be not accepted. To solve this problem, new step was invented to be added to FLASA algorithm which is called "Master Words".

5.4.2   Master Words Step

As shown in figure 5.3 there was a gab between the two successful methods namely; Centroid and MOM in marking question number 7. Instead of having a close results between the instructor method and the FLASA methods the marks achieved by FLASA was not accepted. Therefore, the gab was marked to be missing one of the words which can't be just counted as weight but its present is crucial to give a mark for the answer.

These words, however, were decided to be called "Master Words" for example (increase, decrease), (dependent and independent) ….etc, after adding this step to our algorithm it is obvious in figure 5.10 that the FLASA methods come back to be success in marking.

This step, will be add as first step of our algorithm, so we will add the following step to our algorithm to have the results of example 3 as in table 5.13.

Step one:

IF   Master Word presented   **THEN**

GO to next step

ELSE

Mark is Zero and **EXIT**

                    END IF

Table 5.14: FLASA method with Master Words step.

Student#	Normal Method	Triangular MF		Trapezoidal MF		instructors	Keyword Method
		Centroid Method	MOM	Centroid Method	MOM		
Std#1	9	8.03	9	7.94	8.51	9	9
Std#2	8	7.86	8.28	7.94	8.51	9	6.75
Std#3	8.5	7.94	8.51	7.94	8.51	9	6.75
Std#4	8.5	7.94	8.51	7.94	8.51	9	6.75
Std#5	7	7.86	8.28	7.94	8.51	8	1.8
Std#6	7.5	8.03	9	7.94	8.51	9	9
Std#7	6	0	0	0	0	0	6.75
Std#8	4.5	6	6.03	6	6.03	5	6
Std#9	5.5	6	6.03	6	6.03	8	3.6
Std#10	6	6.74	6.03	6.66	6.03	8	1.8
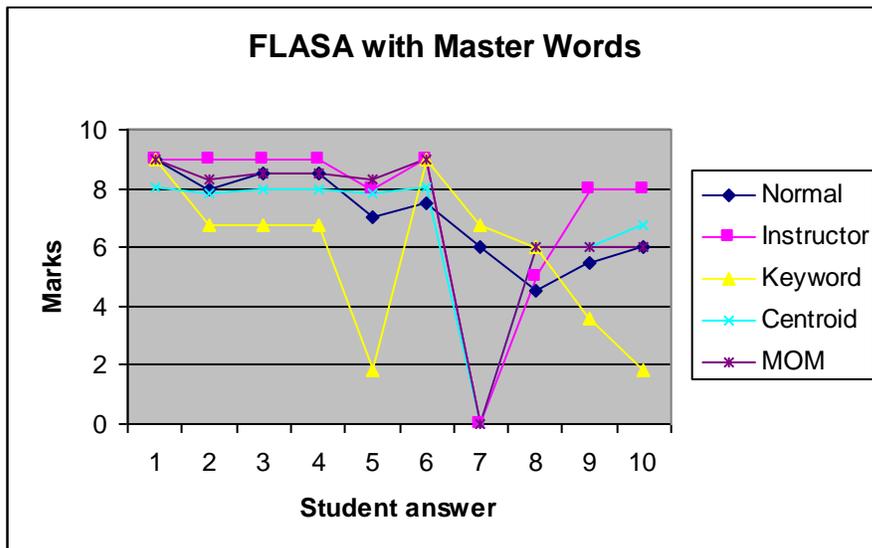
Figure 5.10:- FLASA method with Master Words step.

5.4.3 Experiments Observations

The performed experiments came up with the following observation:

1. Any change in the results of the Rule Base condition will make difference in the final results of the answer marks.

2.   In the first method, the distribution of inputs weights of words will not make any difference in the final results. But in our algorithm "FLASA", there are a dependency between the weights distribution and the final results of the answers marks. These dependency making differences in the final result of the mark, as we see in table 5.14. Table 5.14 shows us the independency between the distribution of the weights and the final marks in the first method, and the dependency between the distribution weights of the words and the final marks in our algorithm.

Table 5.15: dependency and independency of distribution weights

I₁	I₂	I₃	Normal	Centroid	MOM
2	2	2	6	5.93	4.5
1	2	3	6	6.95	6.84
1	2.5	2.5	6	7.86	8.28
.5	2.5	3	6	7.94	8.51
1.5	1.5	3	6	6.83	7.02
1.5	2.5	2	6	6.95	6.84

Table 5.12 also shows the distribution of the input weights are effect the final marks in FLASA algorithm. this is because we works with fuzzy terms like Low, Middle and High instead of number as an input, and the output is fuzzy terms like Zero, Low, Middle, High and Full which denotes from range zero mark to full mark.

5.5 FLASA Features and Advantages.

The main advantages of FLASA include:

1.   FLASA does not depend on references texts, but depends on the main words and their synonyms.

2.   Final mark does not depend on the appearance of the words in the references text, but depends on the weights of these words in the answers key. The orders of the input words in the Rule Base conditions are not important, as we have discussed early in this chapter.

3.   Flexibility: FLASA is very flexible since the final results depend on many parameters in FLASA algorithm, like the weights distributions of the main words, the results of each rule in the rules base, membership functions for output fuzzy set, and the defuzzification methods used to extract the output number.

4.   Students can get instant and detailed feedback about their works, which improves the situation with many instructors that return few or little comments to their students, as shown in appendix B.

5.   Another important advantage of FLASA it can be easily extended and improved by integrating it with other techniques and resources such as thesauri or parsers as MSW dictionary, or any internet dictionary.

5.6 FLASA Disadvantages and Constraints.

FLASA Disadvantages are.

•   The main disadvantage of FLASA is the dependency on the main words number in the key answer, when the number of main words increased; this means that the number of rules in the rules base will be increased by using the following Expression:

Number of rules $= 3^X$

Where x is the number of main words. Thus the net results of this rules number are huge. So we need in one way or another to minimize the number of rules.

- The crucial factors affecting the FLASA performance are the number and quality of the reference similar words used.

FLASA Constraints are.

- It's just for *simple closed questions* such as definitions which have limited numbers of acceptable answers.

- It does not deal with type of questions where the order of main words in key answer is important, like describing behavior of some thing, for example, describe what Ali was doing?, the key answer is " Ali was eating apple". In this case the order of main words which are "Ali, eating, apple", so we could not say that "apple eating Ali".

- It does not deal with type of questions that have lexical or grammatical problems.

- It does not recognize negation if appears in the question.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

The first conclusion that can be drawn from this work is that many universities and companies all over the world are taking an increasingly interest in developing systems to improve the quality of the evaluation task with the aid of computers. Thus to help both students and teachers. Students, because they have given immediate and detailed feedback. Teachers, by cutting-down of the time-consumed in scoring task by just having to check the answers that the computer could not handle or by supervising the process, "letting the hardest work to the computer".

On the other hand, although it is well known that just testing by **MCQs** "Multiple Choice Questions" or fill-in-the-blank questions is not enough to measure the student knowledge, since it depends at most on the chance, but it need less time in evaluation. The appearance of new automated evaluation techniques in this area will open the way for the instructors to choose any type of questions need without concerning the time of evaluation. Moreover, new types of assessments are devised (e.g. face-to-face assessment, alternative assessment, peer assessment or portfolio assessment) enlarging the assessment possibilities.

In particular, we have focused on the automated methods of free text answers both for summative and formative evaluation. This type of evaluation is interesting since it requires students to remember the lesson concepts which making them to compose a writing answer

using their own expressions. Incidentally, it is in the previous statement where the core idea and one of the main problems of automated evaluation of free text answers are exposed: the core idea is to evaluate students emulating human teachers and thus, teachers are asked to write some reference texts or reference texts are automatically retrieved from other sources. Then students' answers are compared against the teachers' ones; and the problem is that students with unusual writing or more creative ideas and their expressions utterly are diverted from the teachers one references and they might be unduly penalized.

From different approach the conclusion that can be drawn are concerning:

1. *The language in which the students answers are written*. The most automated short-text answers systems are limited to the English language; a movement to task into account more languages is being started. It is clear that the complexity of being able to assess texts written in other languages depends on the technique used in the system. FLASA algorithm is language independent, we testing it by using Arabic language as a student's answers.

2. *The suitable domains to assess*, automated short-text answers system are not to evaluate general opinion or mathematical questions, just the opposite qualitative domains such as biology, psychology or history.

3. *What teachers expect from an automated short-text answers system*, it can be highlighted the reliability of the system, that is, its scores should not be too different from the scores that they think.

4. ***What students expect***, it can be highlighted the feedback. They do not want to wait weeks to be finally given just a single score without any justification.

5. ***What should be assessed,*** the general opinion is that both the content and the style is important.

6. ***The results***, the correlation value should be above the 85% for all systems independently from the technique that is being employed.

7. ***The applications***, most of the existing system started as academic products and afterwards have become commercially available. Hence, their authors only provide limited demos on the web. The exception is BETSY that is freely downloadable.

Regarding our approach, we have presented a new use of the Fuzzy Logic "FL" algorithm, for evaluating students' answers "short-text answers" with very promising results (as high as 80% correlation) in comparison with instructors evaluation that has lead us to new research lines:

- To prove that FLASA is a valid approach: by comparing results of FLASA with results of instructors and other methods including keyword ones.

- To build a completely new automated of short-texts answers system based on FLASA: FLASA is not to be used as a stand-alone application, just an internal module to improve the vocabulary analysis or as the basis for adding more complex modules that use NLP.

6.2 Future work

We envision the following open lines for near future work (to be accomplished in the order exposed):

1. **Generate a general system that works and deals with a wide range of n number of key words.**

2. **Extend the FLASA algorithm with more advanced linguistic processing modules**: For instance, a spelling checker, to remove typing mistakes; a rhetorical analyzer, to discover the internal structure of the answer; or an anaphora resolution module, to take into account all words used by the student ignoring the backwards references and making everything explicit.

3. **Extend the FLASA algorithm with more different types of questions**: They would be extremely worthwhile providing the necessary information to improve the results with open answer questions and essay questions.

4. **Discuss the importance of other membership functions, rules, etc.**

Appendix A

Glossary

ATM	Automated Text Marker
BETSY	Bayesian Essay Test Scoring sYstem
Bleu	Bilingual Evaluation Understudy
BM	Bernoulli Model
BP	Brevity Penalty
CAA	Computer Assisted Assessment
DOM	Degree Of Membership
DSS	Decision Support System
ERB	Evaluating Responses with Bleu
ETS	Educational Testing Service
FL	Fuzzy Logic
FLASA	**F**uzzy **L**ogic in **Auto-marking S**hort text **A**nswers
FSM	Finite State Machine
GMAT	Graduate Management Admissions Test
IE	Information Extraction
IEA	Intelligent Essay Assessor
IEMS	Intelligent Essay Marking System
IR	Information Retrieval

ITAL	Interactive Technologies in Assessment and Learning
LSA	Latent Semantic Analysis
MBM	Multivariate Bernoulli Model
MBP	Modified Brevity Penalty
MCQ	Multiple Choice Question
MF	Membership Function
MISOS	Multiple Input Single Output System
MNLP	Microsoft Natural Language Process
MV	Membership Value
NE	Named Entity
NLP	Natural Language Processing
NP	Noun Phrase
PEG	Project Essay Grader
RSA	Rhetorical Structure Analysis
SSA	Syntactic Structure Analysis
SVD	Singular Value Decomposition
TCA	Topical Content Analysis
TOEFL	Test of English as a Foreign Language
UCLES	Unit of the University of Cambridge Local Examinations Syndicate

VG   Verb Group

VLE   Virtual Learning Environment

Appendix B

Technical details

FLASA has been programmed in oracle software using Oracle 8i as a database engine and developer6i as an interface design. This because it can be ported across operative systems without any further modification of the code and it can be viewed thought the internet which means that we can have an on-line version of this method. This is may be in the second version of this system. This is the first version of FLASA, is a prototype of our system, we can introduce some screen used in this system.
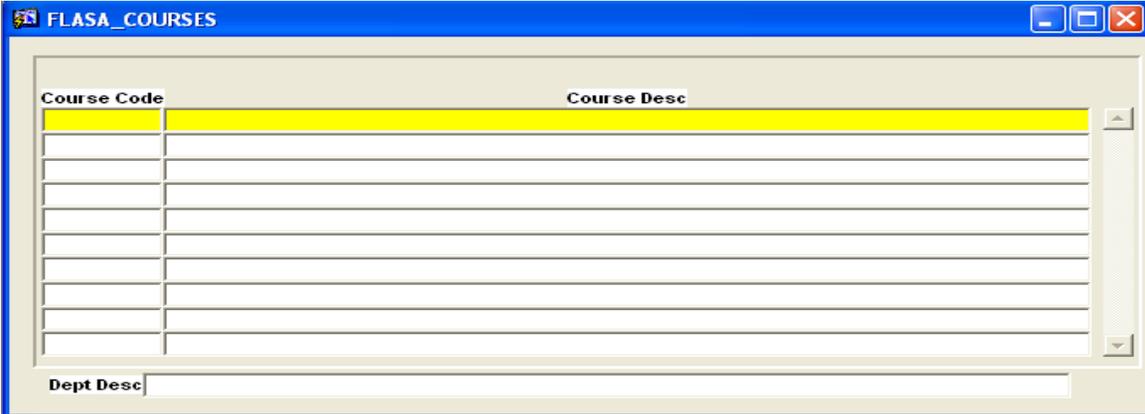
Figure B.1: Define the courses in the university.

Figure B.2: Define the section of the course.
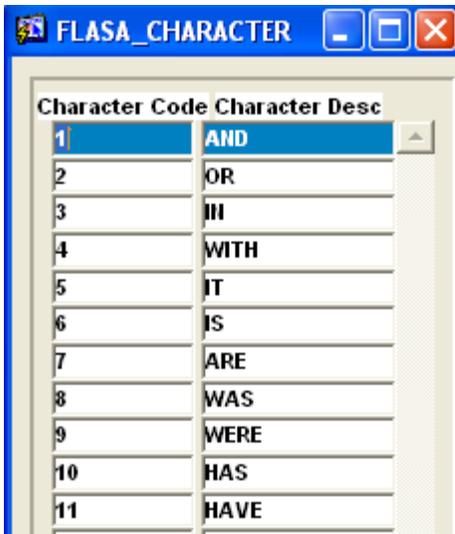
Figure B.3: Define all special character and all stop words to be used in the system.
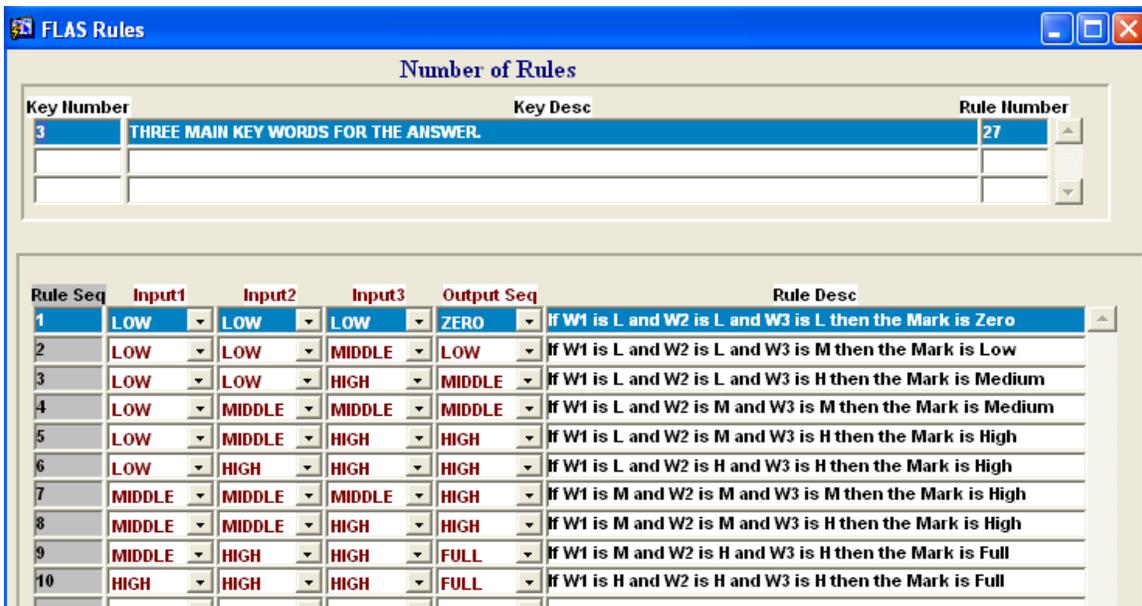
Figure B.4: Define the number of inputs and their related Rule Base.

These rules may define from the instructors or by the system and the instructor can make any change.
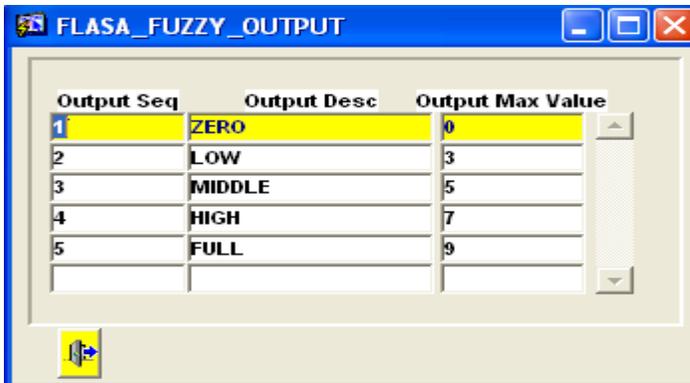
Figure B.5: define the output Fuzzy Set.
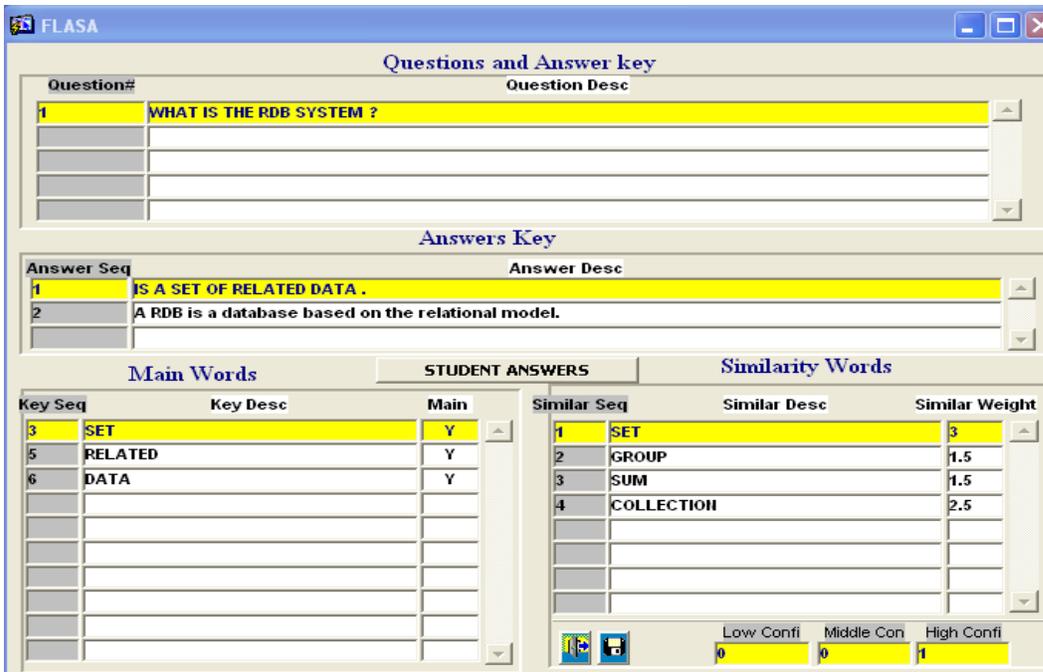
Figure B.6: Define the questions and their answers key.

Here after we define the questions and their answers key, the system will be extract the main words from the answer key by using NLP techniques with help from the screen in figure B.3. And then the system will be define all similarity words using data in table FLASA_SIMILARITY_WORDS or by using MNLP " Microsoft Natural Language Process"

dictionary, then the system or the instructor will put the weight for each one of the similarity words defined.
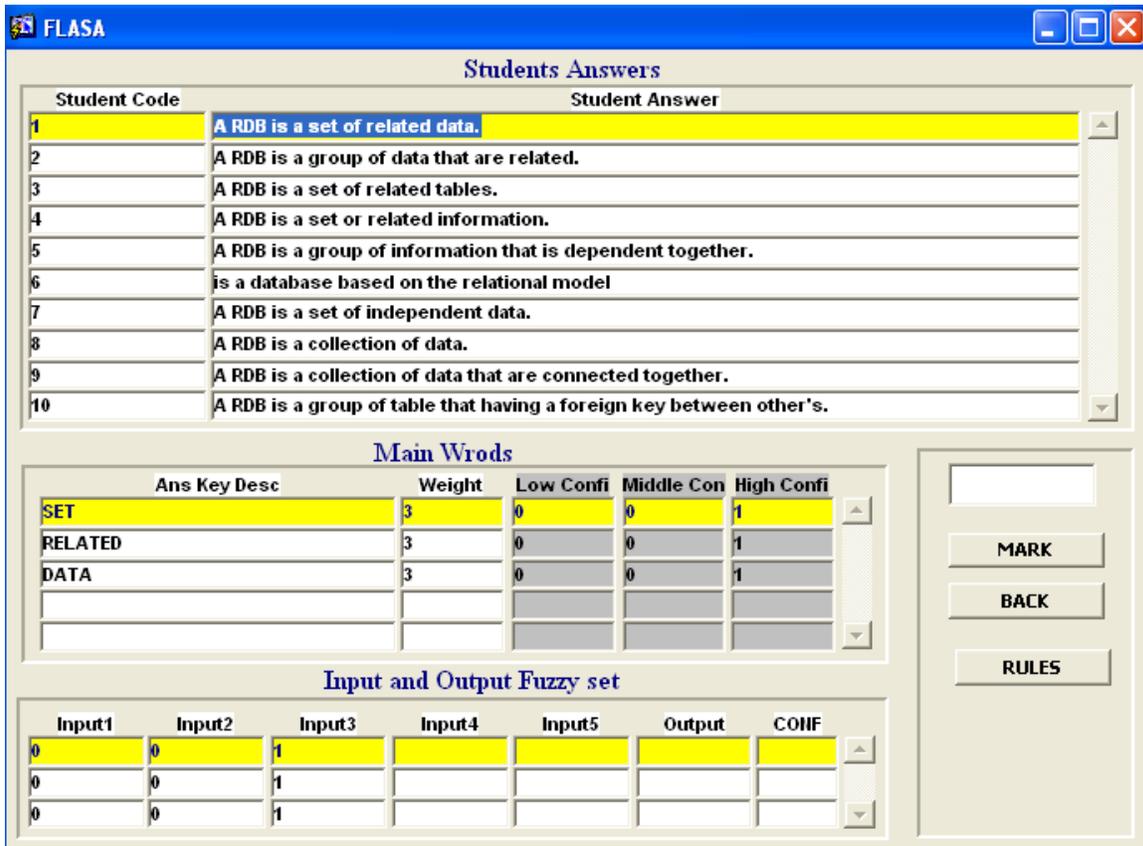
Figure B.7: Define the answers.

Here, the system loaded the students answer from special file or from the internet, and then the system will be extract all main words in the student's answers, and then define the weight for each one of the main words in the answers. This can be dining by keywords method which is the Normal Stage in our FLASA.

After we define the main words and their weight the system will be translate to the FL stage, which is convert each main words in the from the student answer to one of the input fuzzy set with confidence, then the system will be extract all rules from the rule base screen which

linking input fuzzy set with output fuzzy set, and after making defuzzification for these rules

we have a crisp number which is the final mark of the answer.

FLASA INSTRUCTORS	
INSTRUCTOR CODE	789
INSTRUCTOR NAME	A
DEPT DESC	A

FLASA COURSES	
COURSE CODE	789
COURSE DESC	A
DEPT DESC	A

FLASA SECTIONS	
INSTRUCTOR CODE	789
COURSE CODE	789
SECTION CODE	789
SECTION DAY	A
SECTION TIME	A
HOLL CODE	789

FLASA NUMBER KEY	
KEY NUMBER	789
KEY DESC	A
RULE NUMBER	789

FLASA RULES BASE	
KEY NUMBER	789
RULE SEQ	789
INPUT1	A
INPUT2	A
INPUT3	A
INPUT4	A
INPUT5	A
RULE DESC	A
OUTPUT SEQ	789

FLASA FUZZY OUTPUT	
OUTPUT SEQ	789
OUTPUT DESC	A
OUTPUT MAX VALUE	789

FLASA QUESTIONS

QUESTION CODE	789
QUESTION DESC	A

FLASA ANSWER KEY

QUESTION CODE	789
ANSWER SEQ	789
ANSWER DESC	A

FLASA ANSWER KEY MAIN WORDS

QUESTION CODE	789
ANSWER SEQ	789
KEY SEQ	789
KEY DESC	A
MAIN KEY	A

FLASA ANS KEY SIM WORDS

QUESTION CODE	789
ANSWER SEQ	789
KEY SEQ	789
SIMILAR SEQ	789
SIMILAR DESC	A
SIMILAR WEIGHT	789
LOW CONF	789
MIDDLE CONF	789
HIGH CONF	789

FLASA STUDENTS ANSWERS

QUESTION CODE	789
STUDENT CODE	789
STUDENT ANSWER	A

FLASA STUDENT ANSWER RULES

QUESTION CODE	789
STUDENT CODE	789
INPUT1	A
INPUT2	A
INPUT3	A
INPUT4	A
INPUT5	A
OUTPUT	A
OUTPUT CONF	789

FLASA STUDENT ANSWER KEY

QUESTION CODE	789
STUDENT CODE	789
ANS KEY SEQ	789
ANS KEY DESC	A
ANS KEY WEIGHT	789
LOW CONFIDENCE	789
MIDDLE CONFIDENCE	789
HIGH CONFIDENCE	789

Figure B.8: Scheme that shows the tables and relationships among them in the database developed to be used with FLASA.

Appendix C

Questionnaires

C.1 Questionnaire 1.

The aim of the first Questionnaire is to define the time taken to evaluate the same question
with different shapes. And what is the favorite types of questions and why.

<div dir="rtl">

بسم الله الرحمن الرحيم

1.   ان الهدف من هذا الاستبيان هو لمعرفة الوقت الذي يستنفذه المدرس في عملية تصليح مختلف الانواع من

الاسئلة و ما هي نوعية الاسئلة المفضلة لدى الاستاذ و لماذا .

الجزء الاول:-

●   ما هي نسبة الوقت المستنفذ لتصليح الامتحانات خلال الفصل من ساعات

التدريس ....

●   ما هو شكل الاسئلة التي تعتقد بأنها تقيم الطالب بشكل جيد . الرجاء ترتيبها حسب الافضلية مع

توضيح مختصر للسبب.

....

....

....

....

●   ما هي طبيعة الاسئلة التي تفضلها و لماذا .

....

....

....

●   ما هو الوقت المستنفذ لتصليح نفس السؤال بعدة أشكال مثل :-

</div>

1. اختيار الاجابة الصحيحة ....

2. صح أم خطأ . ...

3. تعبئة الفراغ . ...

4. إجابة قصيرة " من سطر الى سطرين " . ...

- هل تأخذ بعين الاعتبار الاخطاء الاملائية في وضع علامة السؤال ؟ الرجاء التوضيح .

- هل تأخذ بعين الاعتبار الاخطاء اللغوية و الخاصة بتركيبة الاجابة من ناحية الاعراب و بناء الجملة ؟ الرجاء التوضيح .

- ما هي الطريقة المثالية التي تتبعها في تصليح الاسئلة التي تحتاج الى إجابة قصيرة " سطر أو سطرين " مثل : عرف ما يلي ، وضح ، علل ، فرق ... الخ ؟

C.2 Questionnaire 2.

The aim of this questionnaire is to making comparison between the instructor evaluation and the FLASA one.

<div dir="rtl" align="center">

بسم الله الرحمن الرحيم

الرجاء تصليح السؤال حسب الاجابة النموذجية مع إعطاء سبب لالية التصليح ، مع الاخذ بعين الاعتبار بأن علامة السؤال هي 9 . الرجاء توضيح الية التصليح في حقل الملاحظات، و إعطاء العلامة في الحقل المخصص لذلك .

</div>

Question#1:- What is the RDB?

The answer key is: the RDB is a <u>set</u> of <u>related</u> <u>data</u>.

The student's answers are:-

Stud	Answers	Grade	Notes
Std#1	A RDB is a <u>set</u> of <u>related</u> <u>data</u>.		
Std#2	A RDB is a <u>group</u> of <u>data</u> that are <u>related</u>.		
Std#3	A RDB is a <u>set</u> of <u>related</u> <u>tables</u>.		
Std#4	A RDB is a <u>set</u> or <u>related</u> <u>information</u>.		
Std#5	A RDB is a <u>group</u> of <u>information</u> that is <u>dependent</u> together.		
Std#6	is a <u>database</u> based on the <u>relational</u> <u>model</u>		
Std#7	A RDB is a <u>set</u> of <u>independent</u> <u>data</u>.		
Std#8	A RDB is a <u>collection</u> of <u>data</u>.		

Std#9	A RDB is a **collection** of **data** that are **connected** together.		
Std#10	A RDB is a **group** of **table** that having a **foreign** **key** between other's.		

References

1. Liew Soot PohBukit Merah Secondary School, Singapore, "Auto Marking System".

2. Benjamin Bloom (ed), Taxonomy of Educational Objectives: Handbook I Cognitive Domain (New York: David McKay Co., 1956).

3. Five Basic Types of Questions on the web at "http://www.uwsp.edu/education /lwilson/ learning/ quest2.htm" at 25/11/2005

4. Bishop, P. (2002), 'Assessment for a purpose', MSOR Connections 2(3).

5. Diana Perez Marin under the supervision of Enrique Alfonseca and Pilar Rodriguez, Automatic evaluation of users' short essays by using statistical and shallow natural language processing techniques".

6. Arthur Woodward, Bruce K. Britton, Learning from Textbooks: Theory and Practice PAGE 57-58.

7. Marti A. Hearst / University of California, Berkeley, The debate on automated essay grading.

8. Saiyam Kohli, Kedar Bhumkar, Vishal Bakshi, Murthy Ganapatibhotla, Apurva Padhye "INDEPENDIENTE - AUTOMATED ESSAY SCORING SYSTEM".

9. Martin Chodorow "Hunter College of the City University of New York ETS, Princeton, NJ " , Jill Burstein "ETS, Princeton, NJ" , Beyond Essay Length: Evaluating e-rater's Performance on TOEFL® Essays.

10. Lin, C. and Hovy, E. (2003), Automatic evaluation of summaries using n-gram co-occurrence statistics, in 'In Proceedings of the Human Technology Conference 2003 (HLT-NAACL-2003)'.

11. E.B. Page and N.S. Petersen, "The Computer Moves into Essay Grading: Updating the Ancient Test," Phi Delta .

12. T. Landauer, and S. Dumais, "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge," *Psychological Rev.*,Vol. 104, 1997.

13. Foltz, P., Laham, D. and Landauer, T. (1999), 'The intelligent essay assessor: Applications to educational technology', Interactive Multimedia Electronic Journal of Computer-Enhanced Learning.

14. Claudia Leacock, Educational Testing Service, Scoring Free-Responses Automatically:A Case Study of a Large-Scale Assessment .

15. Callear, D., Jerrams-Smith, J. and Soh, V. (2001b), Caa of short non-mcq answers, in 'Proccedings of the 5th International Computer Assissted Assessment conference'.

16. Rudner, L. and Liang, T. (2002), Automated essay scoring using bayes' theorem, in 'Proceedings of the annual meeting of the National Council on Measurement in Education'.

17. McCallum, A. and Nigam, K. (1998), A comparison of event models for naive bayes text classification, in 'AAAI-98 Workshop on Learning for Text Categorization'.

18. Mitchell, T. (1997), Machine Learning, WCB/McGraw-Hill.

19. Sukkarieh, J. Z., Pulman, S. G. & Raikes, N. Auto-marking: using computational linguistics to score short, free text responses. Paper presented at the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK. October 2003.

20. Appelt, D. and Israel, D. (1999) Introduction to Information Extraction Technology. IJCAI 99 Tutorial.

21. McCallum, A. and Nigam, K. (1998), A comparison of event models for naive bayes text classification, in 'AAAI-98 Workshop on Learning for Text Categorization'.

22. Papineni, K., Roukos, S., Ward, T. and Zhu, W. (2001), Bleu: a method for automatic evaluation of machine translation, Research report, IBM.[34] {22}

23. William Siler, PhD , Birmingham, AL 35217, USA , "BUILDING FUZZY EXPERT SYSTEMS".

24. Sujit Nath Pant , Keith E. Holbert , Fuzzy Logic in Decision Making and Signal Processing, Fuzzy Logic in Decision Making and Signal Processing, 2004.

25. Allen Bonde, GTE Government Systems Corp ," FUZZY LOGIC BASICS".

26. Steven D. Kaehler, the Boeing Company in Seattle WA, Fuzzy Logic Tutorial, FUZZY LOGIC – AN INTRODUCTION, 1993.

27. Brigette Krantz , "A "Crisp" Introduction to Fuzzy Logic" .

28. Zadeh, L. A., "Making Computers Think Like People", *IEEE Spectrum*, 8, 1984.

29. Martin Hellmann, March 2001," Fuzzy Logic Introduction"

30. Thomas Sowell , FUZZY LOGIC FOR "JUST PLAIN FOLKS" ,

31. George J. Klir/Bo Yuan, Fuzzy Sets and Fuzzy Logic, Theory and Applications.

32. Fuzzy System.Located on the web at "http://datamining.ihe.nl/research/fuzzy.htm", at 25-11-2005.

33. Eral Cox, The Fuzzy Systems Hand book. .

34. VanLehn, K., Jordan, P., Ros´e, C. and Group, T. N. L. T. (2002), The architecture of why2-atlas: a coach for qualitative physics essay writing, in 'Proceedings of the Intelligent Tutoring Systems Conference'.

35. Valenti, S., Neri, F. and Cucchiarelli, A. (2003), 'An overview of current research on automated essay grading', Journal of Information Technology Education 2.

36. L. A. Zadeh, chapter "Fuzzy Sets," in *Information and Control*, Vol. 8, 1965.

37. MUC7 (1998), Proceedings of the 7th Message Understanding Conference (MUC-7), Morgan Kaufman. Nejdl, W., Wolf, B. and Qu, C. (2002), Edutella: A p2p networking infrastructure based on rdf, in 'Proceedings of the 11th World Wide Web Conference'.

38. N. MacDonald et al., "The Writer's Workbench: Computer Aids for Text Analysis," *IEEE Trans. Comm.*,Vol. COM-30, No. 1, 1982.

39. Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., Bradenharder, L. and Harris, M. D. (1998), Automated scoring using a hybrid feature identification technique, in 'Proceedings of the Annual Meeting of the Association of Computational Linguistics'.

40. Callear, D., Jerrams-Smith, J. and Soh, V. (2001a), Bridging gaps in computerised assessment of texts, in 'Proceedings of the IEEE International Conference on Advanced Learning Techniques (ICALT'01)'.

41. Leacock, C. (2004), 'Scoring free-responses automatically: A case study of a large-scale assessment. *Examens, 1*(3).

42. Cowie, J. and Lehnert, W. (1996), 'Information extraction', In Communications of the ACM , 39 (1).

43. Mitchell, T., Russell, T., Broomhead, P. and Aldridge, N. (2002), 'Towards robust computerised marking of free-text responses'.

44. Ming, Y., Mikhailov, A. and Kuan, T. (2000), 'Intelligent essay marking system', Learners Together .

45. Callear, D., Jerrams-Smith, J. and Soh, V. (2001b), Caa of short non-mcq answers, in 'Proccedings of the 5th International Computer Assissted Assessment conference'.

46. P´erez, D., Alfonseca, E. and Rodr´ıguez, P. (2004b), Upper bounds of the bleu algorithm applied to assessing student essays, in 'Proceedings of the 30th International Association for Educational Assessment (IAEA) Conference'.

47. Olsen, M. B. October (1998), 'Translating english and mandarin verbs with argument structure' (MIS) matches Using LCS Representation.

48. Dorr, B. (1994), 'Machine translation divergences: a formal description and proposed solution', Computational Linguistics archive, Volume 20, Issue 4.

49. Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A. (1990), 'Indexing by latent semantic analysis', Journal of the American Society of Information Science.

50. Dessus, P., Lemaire, B. and Vernier, A. (2000), Free text assessment in a virtual campus, in 'Proceedings of the 3rd International Conference on Human System Learning'.

51. Burstein, J., Leacock, C. and Swartz, R. (2001), Automated evaluation of essays and short answers, in 'Proceedings of the International CAA Conference'.

52. Abney, S. (1996), Part-of-speech tagging and partial parsing, Dordrecht: Kluwer.

53. Marcu, D. (2000), 'The theory and practice of discourse parsing and summarization', The MIT Press.

54. Wresch, W. (1993), 'The imminence of grading essays by computer–25 years later', Computers and composition.

55. Burstein, J. and Chodorow, M. (1999), 'Automated scoring for non-native english speakers', Workshop on Computer-Mediated Language Assessment and Evaluation of Natural Language Processing.

56. Mitchell, T., Aldridge, N., Williamson, W. and Broomhead, P. (2003), Computer based testing of medial knowledge, in 'Proceedings of the 7th Computer Assisted Assessment Conference'.