

A Fuzzy Reasoning Database Question Answering System

Stamatis Vassiliadis, *Senior Member, IEEE*, George Triantafyllos, and Walid Kobrosly, *Member, IEEE*

Abstract—The present paper describes a question answering system based on fuzzy logic. The proposed system provides the capability to assess whether a database contains information pertinent to a subject of interest by evaluating each comment in the database via a fuzzy evaluator that attributes a fuzzy membership value indicating its relationship to such subject. An assessment is provided for the database as a whole regarding its pertinence to the subject of interest, and consequently comments that are considered irrelevant to the subject may be discarded. The system has been developed for the examination of databases that were created during the development of the IBM 4381 computer systems, for bookkeeping purposes, to assess whether such databases contain information pertinent to the functional changes that occurred during the development cycle. The system, however, can be applied with minimal changes to a variety of circumstances, provided that the fundamental assumptions for the development of the membership functions are respected in the new application. Its applicability, without modifications, assuming the same subject of interest, is granted for databases comprising similar characteristics to that of the original database for which the system has been developed.

Index Terms—Approximate reasoning, fuzzy reasoning, fuzzy logic, fuzzy sets, fuzzy systems, fuzzy measure, membership functions, question answering system, and database evaluation

I. INTRODUCTION

DURING the different design, test, and release phases of the development of a computer system, a number of databases, in the form of libraries, are developed and maintained for a variety of purposes, such as error tracking and bookkeeping. To understand and improve the development process, previously developed databases may be used at a later date as representatives of the entire process, or as part of the process, and they may be analyzed with the intent of developing algorithms and tools for future use. If a database has been developed with a particular purpose in mind, then it can be used for future studies in its entirety, because the objective of the database was specified a priori. For example, if a database has been developed to report the functional errors discovered during the hardware design of a system, then such a database can be used at the end of the development in its

entirety for a study concerning functional errors in hardware design.

However, on a number of occasions, it may be the case that a developed database needs to be used for a different purpose than originally anticipated. Such a necessity may arise for a variety of reasons, including unanticipated studies. For example, a database that is created from the beginning of the development cycle for bookkeeping purposes, contains information regarding the history of the design, and possibly it may contain information related to functional testing studies if it is assumed that the database is accessed for routine bookkeeping functions and when a functional error is discovered and corrected. Additionally, assuming its applicability is granted, such a database may be considered more representative for functional error studies than an error-tracking library if the latter has been developed at the integration phase of the system development.

When a database is suspected of containing information pertinent to a process, it is possible to presume that the entire database is pertinent to the intended application. An example of this can be found in [1]. However, this may not be a good choice in most of the circumstances, because such a database was not developed to accommodate the application of interest, and it may result in erroneous conclusions regarding the development process. In essence, it is advisable to investigate the suspicion regarding the relevancy of the database to an intended application. In order to assess whether a database contains information pertinent to a subject of interest, it is desirable to develop a tool that provides the capability to assess the validity of the decision, and, when it is found that the library is pertinent to the subject of interest, to exclude irrelevant library entries.

Given that databases that emerge during the development of a computer system generally contain comments, the previous database issues can be addressed with the examination of such comments. A validation methodology can be developed by using probability theory; however, this approach may not be the most appropriate, because the validation of the database must be carried out from the comments of the database, which are written in a natural language, through the use of some form of common reasoning. A consequence of the previous statements is that a probability approach may not be the most appropriate for this type of application [2], for the following reasons.

- An attempt to reduce the nonspecificity or fuzziness inherent to the natural language descriptions may be unwarranted.

Manuscript received June 12, 1991; revised June 8, 1992, and October 2, 1992.

S. Vassiliadis is with the Department of Electrical Engineering, Delft University of Technology, 2628 CD Delft, the Netherlands; e-mail: stamatis@duteca.et.tudelft.nl.

G. Triantafyllos is with the Poughkeepsie Engineering Laboratory, IBM Corp., Poughkeepsie, NY USA; e-mail: triantaf@vnet.ibm.com

W. Kobrosly is with the Power PC Division, IBM Corp., Austin, TX, USA; e-mail: kobrosly@uasvm6.vnet.ibm.com.

IEEE Log Number 9213328.

- It is not clear how probability theory could effectively describe and manipulate the great variety of descriptions or rules that are possible in natural language.

A second approach to the validation of a commented database is to develop a natural language question answering system. Although it maybe entirely possible to develop such a system, possibly with the use of fuzzy relations [3], such a solution may not be the most advantageous for a number of reasons, including the following.

- A computer system may be developed in more than one country, and consequently the comments of the databases maybe written in more than one natural language, implying that more than one system needs to be developed. An example of this is the IBM 9370 computer systems, which were developed in a number of countries, including the United States and Germany [4].
- Verifying the validity of a database is not the final purpose of the study. The development of natural language question answering systems may require substantial effort, and possibly, if the database is not appropriate to the application, it will need to be discarded. The concern here is not so much with the possible solution to the problem, but rather with the implementation and development efforts, especially when it may be assessed at a later phase of the project that the database does not pertain to the intended purpose of an application.
- A natural language question answering system may not be applicable to different databases and/or to different types of investigations in its entirety; consequently, at least a portion of the question answering system needs to be modified and/or expanded to reflect the wanted application.

Yet another way to determine the validity of such databases is to treat each record of the database as a document and use an information retrieval system to extract the records pertaining to a particular subject of interest. Several such systems have been developed in recent years [5]. The success of a retrieval system in the type of the databases that we consider may not be warranted for several reasons, e.g., deterministic answer to a particular query, insensitivity to the order in which the keywords appear in the document.

A consequence of the previous discussion is that it is of interest to develop a tool that requires a negligible development effort, that expedites and facilitates the evaluation of a commented database with minimal effort, and that can be applied in a variety of circumstances with minimum additional development effort. Such a tool will most certainly allow for more time to be exerted on the analysis of the database rather than the assessment of its applicability. Moreover, if it is assessed that the system is not accurate enough to guarantee a reasonable exclusion of comments, the system could then be used as an indicator of compliance of a database to a prespecified application, and its capabilities may be extended with the use of a natural language question answering system to further investigate the relevancy of the database. The implication is that a natural language question answering system needs to be implemented only in the case when a more

accurate analysis is required. Consequently, the development of such a system will take place when it is needed, rather than a priori.

In this paper, we present a question answering system based on fuzzy logic. The mathematical foundations and in-depth descriptions of fuzzy set theories and fuzzy logic systems can be found in [3], [6]–[9]. The tool provides a quick assessment of the applicability of a database to a specified universe of discussion, and the exclusion of the irrelevant comments of the database.

In the sections to follow, we provide a brief description and an intuitive justification and reasoning for the development of the fuzzy question answering system. The concept of degree of confidence as it relates to words and comments within a database is then, formally defined and formulated. In the subsequent discussion, the fuzzy evaluator algorithm is described, and its capabilities are discussed. The last section contains an evaluation of the performance of the proposed system, followed by some concluding remarks.

II. DESCRIPTION AND INTUITIVE JUSTIFICATION OF THE DATABASE QUESTION ANSWERING SYSTEM

The fuzzy question answering system, presented in this section, has been developed as a means of evaluating commented databases suspected of containing information pertinent to functional testing (i.e., source of number of changes that were made during the development of the system). A brief description of the system follows.

First, all comments are examined and a list is created containing all the unique words present in the database. Subsequently, a list with the most relevant words considered to potentially describe association/disassociation to the subject of interest, i.e., functional changes, together with a degree of confidence for every word, is generated describing the perception of qualified personnel. Finally, a modified database containing comments with relevant words is analyzed with a fuzzy evaluator that assesses the database pertinence to the subject of interest.

Although the design of the system has been targeted to a particular subject of interest, it can be applied with small modifications in a variety of applications, provided that the fundamental assumptions underlying its implementation are granted. The section of the system that requires modifications pertains to the selection of the words that are pertinent to a particular application, and to the interviewing of expert personnel in order to attribute the membership grade values to the relevant words.

The first obvious observation is that the system does not deploy syntactic and semantic rules, and it operates on modified comments that contain partial description of the original comments. A key question that arose before the development of this tool and the verification of its validity is why such a system operating on partial description of comments (implying, among other things, loss of structure and word association and meaning) may be considered promising. An intuitive justification for the possible goodness of this approach is the following.

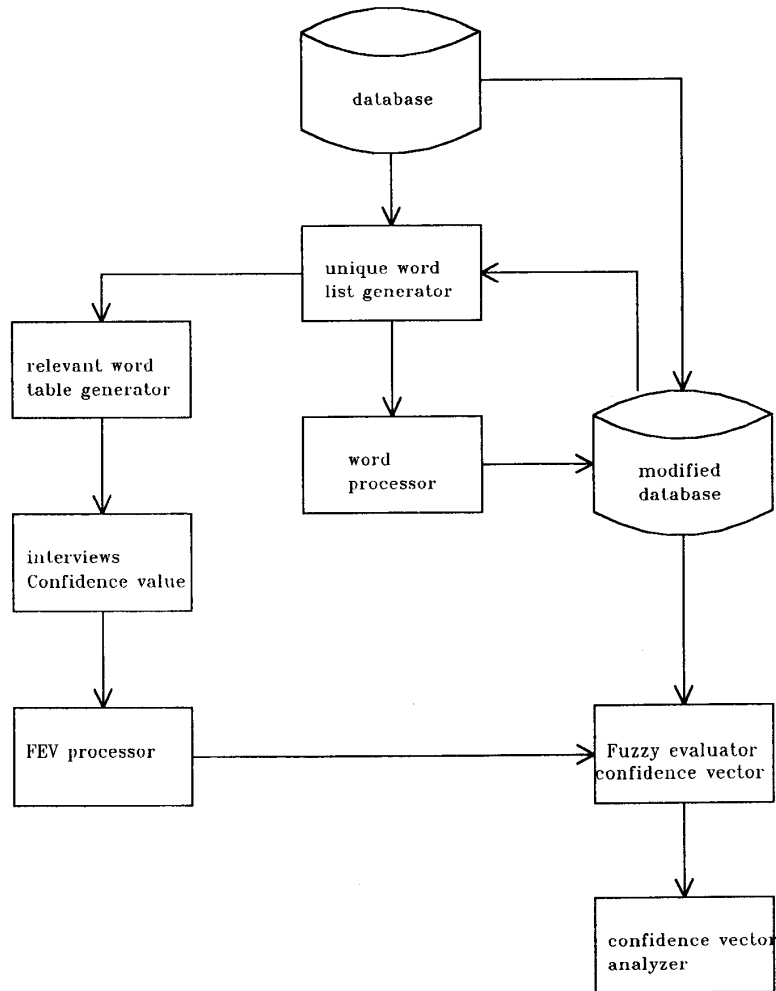


Fig. 1. The Question Answering System.

- Comments are not written at random, and reflect the intentions of the commentator.
- Comments are short; consequently, they most likely reflect what has happened or what has been accomplished with very few key words.
- Most likely, comments reflect what happened or what was accomplished, rather than what did not happen, or was not accomplished; i.e., most of the times there are no negative statements.
- A comment will most likely reflect a single, or a plurality of, relevant/irrelevant actions during a database access, rather than a plurality of irrelevant actions intermingled with relevant actions. (The principle of a single, or a plurality of, relevant/irrelevant actions is referred to as the single action principle in the rest of the presentation.)

Consequences of the previous statements are that the intentions of a comment can be captured with few words, and that the actions associated with a database access can be

extrapolated without having to comprehend the exact contents of a comment. Obviously, the precise actions cannot be associated with a comment (such a determination may require the context in which the words appear among other things); however, this may not be a requirement when the intention of the examination is not the determination of a particular cause, but rather the relationship of a comment to a subject of interest. For example, in examining a bookkeeping library in order to establish its pertinence to functional testing and to extrapolate the comments that are related to this subject, it is required to establish whether the comments in this library are related to functional changes in general, rather than the exact nature of changes that might have occurred in a particular piece of logic or microcode.

The previous discussion leaves open the question of having to find the precise cause of a change. To develop algorithms regarding system testing, precise causes of the changes maybe considered irrelevant if it is of interest to identify, e.g.,

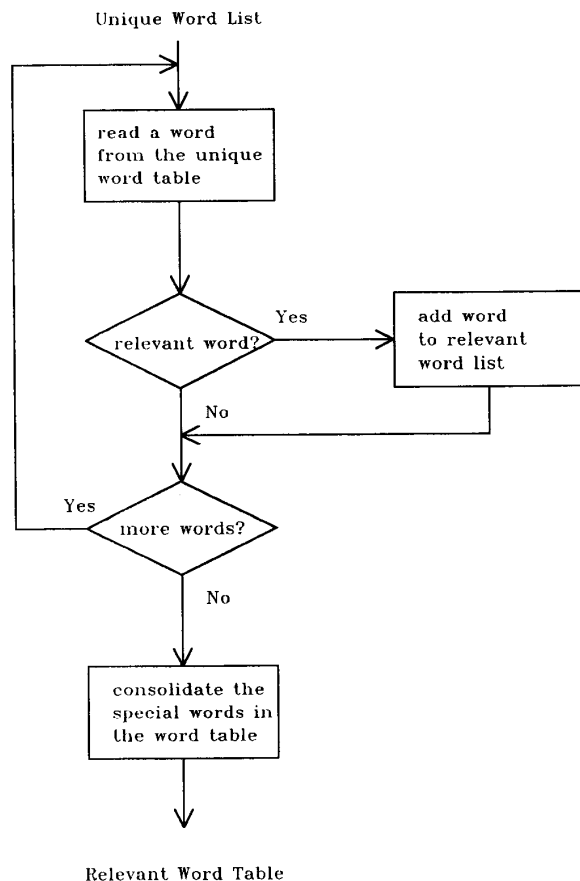


Fig. 2. The Word Processor.

the number of errors, when they occur, and the part of the machine where they occurred, rather than to establish whether the changes were due, e.g., to the wrong polarity of a signal or the wrong implementation of microinstructions. Consequently, the drawback of the proposed system may not be of any importance to the overall results. However, though the example application may not require to establish the exact reason of an action, there may be other circumstances in which such a capability is highly desirable. In such a case, the proposed system can be used only as an indicator by establishing whether a database is relevant to a subject of interest, and determining which entries can be discarded. Consequently, it is required to develop a more powerful tool, such as a natural language question answering system, which can be applied for further investigations.

The overall design of the proposed system is shown in Fig. 1. The Question Answering System comprises the following:

- The Unique Word Generator,
- The Relevant Word Table,
- Interviews,
- The FEV Processor,
- The Word Processor,

- The Modified Database,
- The Fuzzy Evaluator, and
- The Result Analyzer.

The Unique Word Generator generates an alphabetically sorted list of all unique words contained in the comments of the database. This list contains legitimate words, misspelled words, and acronyms that apply to a particular project. The Word Processor, whose operation is described in Fig. 2, requires the manual processing of the words and the automatic examination of the database. During this manual process, the list of the unique words generated by the Unique Word Generator is analyzed in order to extract and place potential relevant words of association/disassociation in a table, which is referred to as the replace table. The replace table contains two columns: The first contains the words as they appear in the database, and the second contains the synonyms corresponding to the words that appeared in the first column. This allows for illegitimate words, such as misspelled words and acronyms, to be properly substituted in the database. For example, if ERROR and FIX are considered relevant words, then EROR can be substituted with ERROR, and FIXED can be substituted with FIX.

During the automatic processing, the records of the original database are read, and each word of every comment in a record is compared with the words in the replace table. All words that match the first column of the replace table are replaced by the words on the second column, and all other words are deleted. Note that if more than one pass of the algorithm is needed to access the entire database, the irrelevant words maybe saved for further processing in the delete table and used to add new entries in the replace table. The reason why more than one pass of the algorithm may be required is the process of choosing the relevant words is a laborious one, and relevant words may be missed when constructing the replace table. To ensure that all relevant words are kept, this step may be performed more than once by reviewing the delete table, which is smaller than the original table, with the unique words and updating the replace table. As a result of this analysis, New comment records are generated, and a second database is established. This database, which is referred to as the modified database, contains the words that are present only in the replace table. Using the Unique Word Generator, and operating on the modified database, the list of all unique words in the modified database is generated, and it is referred to as the relevant word table. Subsequently, the relevant word table is distributed to personnel who are considered to be experts in the subject under investigation, and possibly to members of the group that created the database. Each person is asked to attribute a confidence value for every word that reflects his or her perception of the usage of that word with respect to the subject of interest. In other words, the confidence value indicates how a person perceives that a particular word is most likely used in the comment field of the database.

After the interview, it is required to establish the Fuzzy Expected Value (FEV) for every word in the relevant word table. Such a task is achieved by the FEV Processor, which reads the responses of the interviews and assigns a membership grade to each word in the relevant word table. The FEV

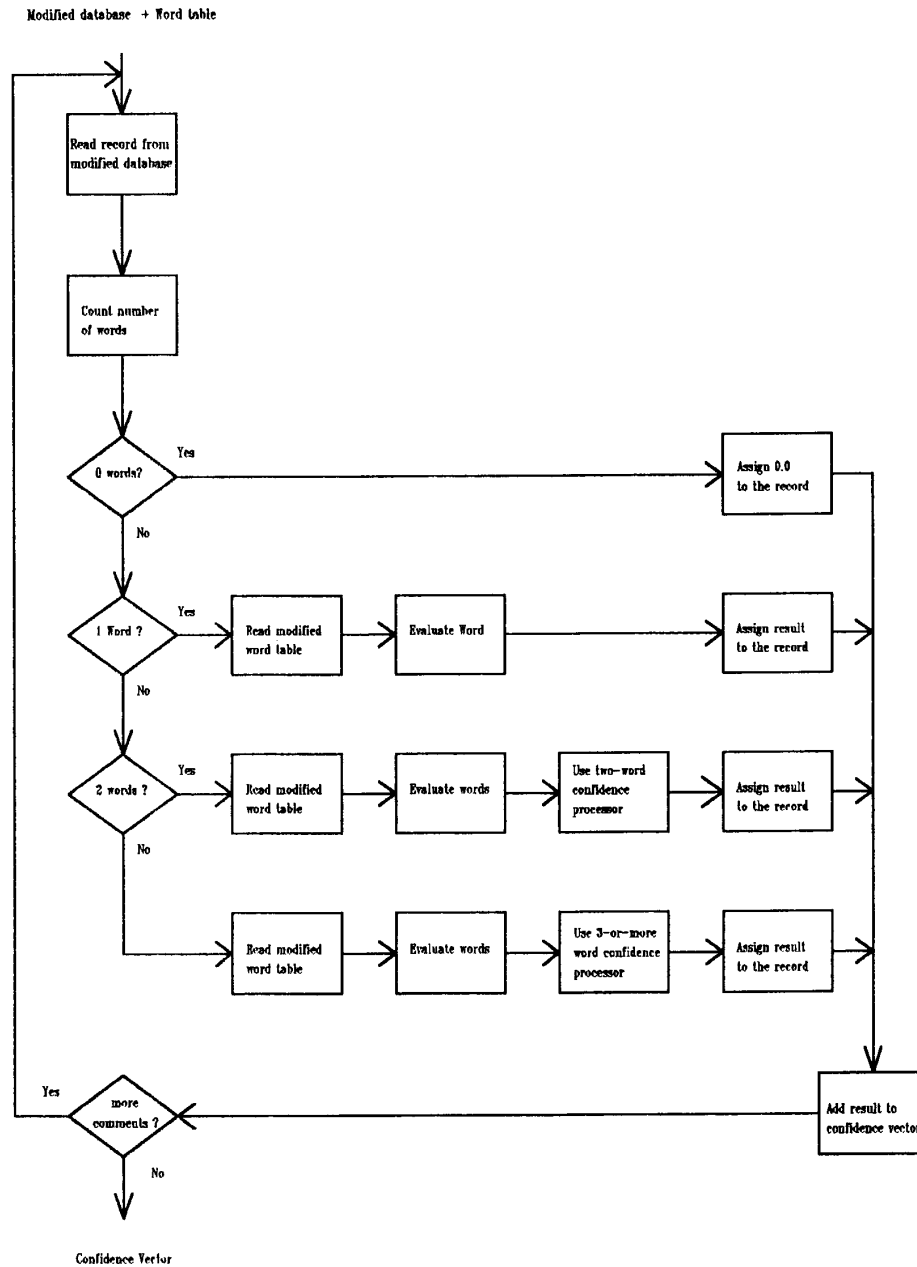


Fig. 3. The Fuzzy Evaluator.

Processor can operate with the use of any of the algorithms proposed by [7], [10]–[12].

The Fuzzy Evaluator operates by analyzing each comment within the modified database in conjunction with the list of words in the relevant word table, as well as the degree of confidence associated with each word. This analysis is based on the number of words within each modified comment record. Each comment is examined as described in Fig. 3. If there are no words associated with a record, then the confidence value of the record is zero. If there is one word, then the confidence

associated with this word is the confidence of the entire record. If there are two words in a comment record, the confidence values of the two words are aggregated based on the following discussion.

For simplicity of exposition, analogous conclusions can be derived for pertinent cases, assuming that a database is suspected to contain information pertinent to functional testing. In essence, assume that the database comments are suspected to report logic changes. Furthermore, assume that the following are the words in the relevant word table.

| | |
|---------|-------|
| ERROR | / 0.8 |
| BUG | / 0.9 |
| FIX | / 0.8 |
| UPDATE | / 0.5 |
| COMMENT | / 0.1 |
| SYNTAX | / 0.1 |

Where X/y indicates the following pair: significant word X with confidence value y . Additionally, assume that there are only two words appearing in the modified database, and that there are the following five comments:

1. < SYNTAX, ERROR >
2. < COMMENT, SYNTAX >
3. < UPDATE, COMMENT >
4. < FIX, UPDATE >
5. < FIX, BUG >

Attributing a degree of confidence to a comment may be interpreted to require the use of an AND operator between the two degrees of confidence of the single words. There are three popular AND operators, namely, the Min operator, the Product operator, and the Bounded Sum operator [13]. Fig. 4 reports the confidence values for all records using the three operators.

The Min operator definition implies that when a pair of words is examined, the confidence associated with the pair needs to be the minimum confidence of the two words. It can be contended that the confidence value 0.1, given by the Min operator to the first comment is close to the mark, or a bit pessimistic, because it indicates that the record does not report a substantial change, that is, if it is assumed that the < SYNTAX ERROR > is a trivial fix. Additionally, under such an assumption, it can be stated that any scheme that produces a confidence value that does not exceed 0.5 (0.5 being the indifference point) will be close to the mark. In other words, an acceptable confidence value for the first comment can be the Min value, or an increase of the Min value, but not to the point that it exceeds 0.5. The confidence value 0.1 given by the Min operator to the second comment may be considered to be overly optimistic, because both words indicate that there is a trivial fix, and consequently, the confidence attributed to the pair < COMMENT SYNTAX > should have been decreased. It should be noted that though both the first and the second comment may be viewed as nonsubstantial changes, the first comment should have a confidence value greater than the second, because of the presence of the word ERROR.

The confidence attributed to the third comment may be considered as either correct for the same reasoning as in comment 1, or a bit overly pessimistic. The confidence attributed to the fourth comment may be considered overly pessimistic, because the fourth comment may refer to an update due to a problem more than an irrelevant update. Consequently, the confidence value should be closer to 0.8 than 0.5. Finally, the confidence value 0.8, attributed to the pair < FIX BUG > may not be considered representative of the comment, because in this case the confidence of a change should be reinforced rather than associated with the lowest value. In other words, the presence of the words "FIX" and "BUG" provides more confidence that a change has been reported than either the confidence of any

| Comment | MIN Operator | PRODUCT Operator | BOUNDED SUM Operator | AVERAGE | Fuzzy Evaluator |
|-------------------|--------------|------------------|----------------------|---------|-----------------|
| <SYNTAX, ERROR> | 0.10 | 0.08 | 0 | 0.45 | 0.27 |
| <COMMENT, SYNTAX> | 0.10 | 0.01 | 0 | 0.10 | 0.03 |
| <UPDATE, COMMENT> | 0.10 | 0.05 | 0 | 0.30 | 0.14 |
| <FIX, UPDATE> | 0.50 | 0.40 | 0.30 | 0.65 | 0.78 |
| <FIX, BUG> | 0.80 | 0.72 | 0.70 | 0.85 | 0.97 |

Fig. 4. Pair confidence values for the AND operators, the average, and the Fuzzy Evaluator.

of the two words. In such a case, the confidence should be greater than at least 0.8, and even greater than 0.9.

The Product operator will produce the confidence value of a pair with the multiplication of the two confidences. The discussion regarding the Min operator also applies to the Product operator with the additional observations that the Product operator performs better than the Min operator for comment 2, and worse for all the other comments. Finally, the Bounded Sum operator will produce the confidence value by $\max [0, (\mu_x + \mu_y - 1)]$, with μ_i denoting the confidence value attributed to word i . Such an operator may be considered overly pessimistic for all comments. It may be contended that the three AND operators may not be appropriate to attribute a confidence value for the given circumstances. This may be attributed to the following observations.

- 1) If the two words are favorably associated with the subject of interest and appear in the same sentence, they most likely reinforce the confidence that the sentence describes the subject of interest. Thus, the resulting confidence should be closer to the maximum, and possibly exceeding the maximum value, when the confidence of a word or both words get closer to a certain association value.
- 2) If the two words are both adversely associated with the subject of interest, then the comment in which these two words appear most likely, does not describe the subject of interest. Thus, the confidence of the two words should be closer to the minimum and possibly less than the minimum value as the two confidence values get closer to a certain disassociation value.
- 3) If two words have the cutoff value, i.e., if both words are neither associated with the subject of interest nor disassociated, then their combined confidence should be the cutoff value itself.
- 4) Excluding the cases of certainty of association and disassociation, which are addressed in the later part of this section, if one word is unfavorable and the other is favorable, then in the case where one word is strongly unfavorable, as in comment 1 in the assumed example, the confidence value of the comment should be closer to the minimum value; otherwise, it should be closer to the maximum value. The previous statement appears to lie on the extreme side; however, there is an intuitive justification in that a word strongly disassociated from the subject of interest paired with a word associated

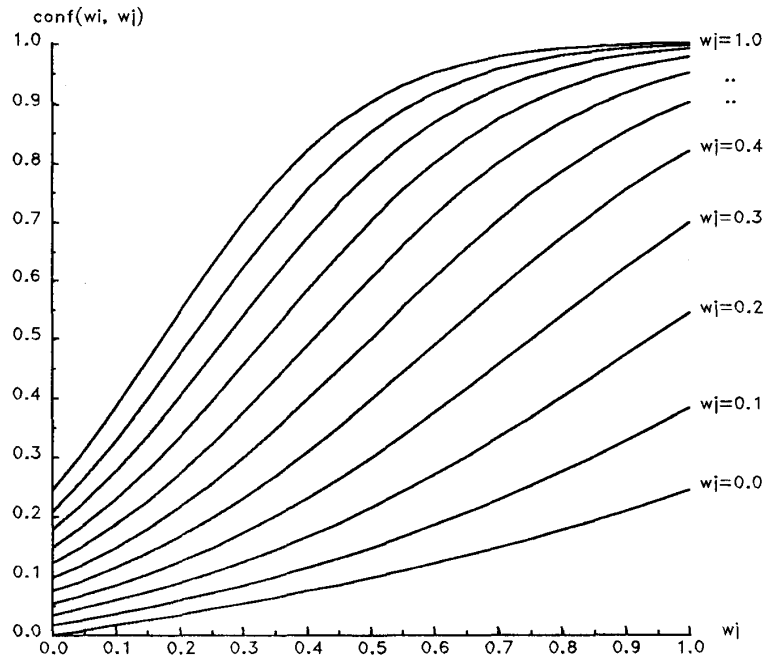


Fig. 5. The membership function for pairs of words.

(even strongly associated) with the subject of interest reinforces somehow the disassociation rather than the association, and the other way around. For example, the words SYNTAX and COMMENT, when paired, e.g., with the words ERROR and BUG, most likely indicate that there is a trivial fix rather than a substantial fix.

The previous discussion indicates that a membership function that can closely reflect the previous four observations needs to be provided. In search of such a membership function, we established that such a function is somehow related to the average of the confidence values attributed to the two words. However, as can be observed in Fig. 4, though the average represents, to a degree, the confidence associated with a comment, it can be used only as a first approximation, and it requires an adjustment to produce a more accurate confidence value. The following discussion describes a membership function that produces an adjusted average that better approximates the previous observations.

Let $\mu_A(w_i, w_j)$ be the membership function attributing a degree of confidence with regard to a subject of interest A for a comment, where w_i and w_j are the confidence values associated with two individual words i and j appearing in the same comment, then $\mu_A(w_i, w_j)$ can be computed by using the following equation:

$$\mu_A(w_i, w_j) = \frac{1}{1 + ce^{-k(w_i w_j - l^2)}}, \quad (1)$$

where k is a constant greater than 0 and $c = \frac{1-m}{m}$, with m being $m = \frac{w_i + w_j}{2}$, i.e., the average confidence of the two words i and j , and l , which is $0 < l < 1$, indicates the following. Words r that have confidence value $w_r > l$

are considered to favorably describe the subject of interest, whereas words with $w_r < l$ are considered to adversely describe the subject of interest, and words with $w_r = l$ are considered to be neutral.

Equation (1), described in Fig. 5 with $l = 0.5, k = 4.5$, is based on the solution of the logistic growth differential equation, and is a modification of the membership functions proposed by M. Kochen and A. Badre in [14], and by P. Zysno in [15]. The following are some of the properties of the membership function.

Assume, for simplicity of notation, that $w_i = x, w_j = y$, and $\mu_A(w_i, w_j) = \mu(x, y)$. Then the membership function can be written as follows:

$$\mu(x, y) = \frac{1}{1 + ce^{-k(xy - l^2)}}.$$

Property 1: If $x = y = 0 \Rightarrow \mu(x, y) = 0$.

Proof:

$$\begin{aligned} \mu(x, y) &= \frac{1}{1 + ce^{-k(xy - l^2)}} = \frac{1}{1 + \frac{1-m}{m} e^{kl^2}} = \\ &= \frac{1}{1 + \frac{1-0}{0} e^{kl^2}} = \frac{1}{1 + \infty} = 0. \end{aligned}$$

Property 2: If x and $y > l \Rightarrow \mu(x, y) > \frac{x+y}{2}$.

Proof: x and $y > l \Rightarrow xy > l^2$. Let $\xi > 0$, and let $\zeta > 1$. Then we have the following:

$$\begin{aligned} \mu(x, y) &= \frac{1}{1 + ce^{-k(xy - l^2)}} = \frac{1}{1 + ce^{-k\xi}} = \frac{1}{1 + \frac{c}{e^{k\xi}}} \\ &= \frac{1}{1 + \frac{c}{\zeta}} = \frac{\zeta}{\zeta + c} = \frac{\zeta}{\zeta + \frac{1-m}{m}} = \frac{\zeta m}{\zeta m + 1 - m}. \end{aligned}$$

It must be that the following conditions exist:

$$\frac{\zeta m}{\zeta m + 1 - m} > m \Rightarrow \zeta m > m(\zeta m + 1 - m) \Rightarrow \zeta > \zeta m + 1 - m \Rightarrow .$$

$$\zeta - \zeta m > 1 - m \Rightarrow \zeta(1 - m) > 1 - m \Rightarrow \zeta > 1.$$

Q.E.D.

Property 3: If x and $y < l \Rightarrow \mu(x, y) < \frac{x+y}{2}$.

Proof: x and $y < l \Rightarrow xy < l^2$. Let $\xi < 0$, $\omega > 0$, and $\zeta > 1$. Then we have the following:

$$\begin{aligned} \mu(x, y) &= \frac{1}{1 + ce^{-k(xy-l^2)}} = \frac{1}{1 + ce^{-k\xi}} = \frac{1}{1 + ce^\omega} \\ &= \frac{1}{1 + c\zeta} = \frac{1}{1 + \frac{1-m}{m}\zeta} = \frac{m}{m + (1-m)\zeta}. \end{aligned}$$

It must be that the following conditions exist:

$$\frac{m}{m + (1-m)\zeta} < m \Rightarrow m < m(m + (1-m)\zeta) \Rightarrow$$

$$1 < m + (1-m)\zeta \Rightarrow 1 - m < (1-m)\zeta \Rightarrow 1 < \zeta.$$

Q.E.D.

Property 4: If $x = y = l \Rightarrow \mu(x, y) = l$.

Proof:

$$\mu(x, y) = \frac{1}{1 + ce^0} = \frac{m}{m + 1 - m} = m = \frac{l + l}{2} = l.$$

Property 5: If x or $y < l$ and $xy > l^2 \Rightarrow \mu(x, y) > \frac{x+y}{2}$.

Proof: Analogous to Property 2.

Property 6: If x or $y < l$ and $xy < l^2 \Rightarrow \mu(x, y) < \frac{x+y}{2}$.

Proof: Analogous to Property 3.

Property 7: If $x = y = 1 \Rightarrow \mu(x, y) = 1$.

Proof: Trivial ($m = 0$ and $\mu(x, y) = 1$).

Properties 2 and 3 reflect observations 1 and 2, because they attribute confidence values always toward the maximum and minimum confidence value of the word pair, respectively. Property 4 closely reflects observation 3. Properties 5 and 6 reflect observation 4 as Fig. 6 suggests. In Fig. 6, it is assumed that the confidence values x and y can take values between 0 and 1 with a 0.1 increment, and it indicates that when a confidence value of a pair of a words is strongly disassociated from the subject of interest (the x values 0.1 and 0.2), then a confidence value attributed to the pair is always closer to the value x rather than the value of y . When they are disassociated, but not "strongly," then for strongly associated values of y , such as 0.9 when $x = 0.3$, the confidence value of the pair becomes closer to the association rather than the disassociation.

Properties 1 and 7 indicate that the boundary conditions are respected; however, the membership function, as described by (1), may not be good enough to guarantee correct values for all possible boundary conditions. In essence, when one of the confidence values, but not both, are different than either 1 or 0, the membership function described in (1) will provide an in-between value, and such a value may not be considered proper. If it is assumed that certainty for association and disassociation

| x Values | Confidence Closer to the y Values |
|----------|-----------------------------------|
| 0.1 | never |
| 0.2 | never |
| 0.3 | if $y > 0.8$ |
| 0.4 | if $y > 0.6$ |

Fig. 6. Properties 5 and 6, pair confidence values.

from the subject of interest is described by the values 1 and 0, respectively, and that the presence of a word with such confidence values, independent of the confidence value of the other word, is prevalent, then (1) will not compute the correct value for a pair of words, and needs to be adjusted to express such a prevalence.

It needs to be noted that the previous discussion leaves open the question of one word having a value equal to 1 and the other equal to 0. If it is assumed that a comment cannot be contradictory, then each word must represent two different scenarios: one that indicates relevance and another indicating irrelevance, implying that in such a case, the resulting confidence value should be equal to 1. In essence, in the assumed example database, if there is a word that indicates that the comment reports a definite change and the other word indicates a definite no-change, then we assume that there are two different actions present in the comment, and that a change is definitely reported. The previous discussion can be incorporated in the two following complementary statements.

- If either w_i or w_j is equal to 1, then $\mu_A(w_i, w_j) = 1$.
- If either w_i or w_j is equal to 0 and the other is different than 1, then $\mu_A(w_i, w_j) = 0$.

Figs. 7 and 8 report two examples of pair membership values using $k = 1$ and $l = 0.5$, and $k = 4.5$ and $l = 0.5$, respectively, that incorporate the two complementary statements.

If the modified database comment contains more than two words, then the confidence value associated with the comment may be computed by the following. If a comment contains n relevant words, its confidence value can be attributed by applying the following formula:

$$\Phi_{i=1}^{n-1} \mu_A[(w_i, w_{i+1})],$$

to the confidence values of the words present in a comment. The operator Phi , for any given i , applies (1) with possible inclusion, if considered appropriate, to the two complementary statements presented previously. The inputs to (1) are the confidence values attributed to words i and $i + 1$, and the final output value of Φ for all i 's between 1 and including $n - 1$ is a set of confidence values. Consequently, a confidence value is attributed to the comment by applying the following algorithm.

- *Step 1:* If there exists at least one element in the set produced by Φ that exceeds a threshold value ρ_0 , and

| x\y | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0.0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 |
| 0.1 | 0.000 | 0.000 | 0.123 | 0.167 | 0.213 | 0.260 | 0.308 | 0.358 | 0.408 | 0.460 | 1.000 |
| 0.2 | 0.000 | 0.123 | 0.169 | 0.216 | 0.266 | 0.317 | 0.369 | 0.423 | 0.478 | 0.533 | 1.000 |
| 0.3 | 0.000 | 0.167 | 0.216 | 0.268 | 0.321 | 0.376 | 0.433 | 0.490 | 0.548 | 0.605 | 1.000 |
| 0.4 | 0.000 | 0.213 | 0.266 | 0.321 | 0.379 | 0.438 | 0.498 | 0.557 | 0.617 | 0.675 | 1.000 |
| 0.5 | 0.000 | 0.260 | 0.317 | 0.376 | 0.438 | 0.500 | 0.562 | 0.624 | 0.683 | 0.740 | 1.000 |
| 0.6 | 0.000 | 0.308 | 0.369 | 0.433 | 0.498 | 0.562 | 0.626 | 0.688 | 0.746 | 0.800 | 1.000 |
| 0.7 | 0.000 | 0.358 | 0.423 | 0.490 | 0.557 | 0.624 | 0.688 | 0.748 | 0.804 | 0.854 | 1.000 |
| 0.8 | 0.000 | 0.408 | 0.478 | 0.548 | 0.617 | 0.683 | 0.746 | 0.804 | 0.855 | 0.901 | 1.000 |
| 0.9 | 0.000 | 0.460 | 0.533 | 0.605 | 0.675 | 0.740 | 0.800 | 0.854 | 0.901 | 0.940 | 1.000 |
| 1.0 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

Fig. 7. The membership grade values for $k = 1$ and $l = 0.5$.

| x\y | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0.0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 |
| 0.1 | 0.000 | 0.036 | 0.059 | 0.085 | 0.115 | 0.148 | 0.186 | 0.229 | 0.276 | 0.327 | 1.000 |
| 0.2 | 0.000 | 0.059 | 0.089 | 0.124 | 0.166 | 0.215 | 0.271 | 0.333 | 0.400 | 0.471 | 1.000 |
| 0.3 | 0.000 | 0.085 | 0.124 | 0.173 | 0.231 | 0.298 | 0.374 | 0.455 | 0.539 | 0.621 | 1.000 |
| 0.4 | 0.000 | 0.115 | 0.166 | 0.231 | 0.308 | 0.395 | 0.489 | 0.583 | 0.673 | 0.753 | 1.000 |
| 0.5 | 0.000 | 0.148 | 0.215 | 0.298 | 0.395 | 0.500 | 0.605 | 0.702 | 0.785 | 0.852 | 1.000 |
| 0.6 | 0.000 | 0.186 | 0.271 | 0.374 | 0.489 | 0.605 | 0.711 | 0.800 | 0.868 | 0.917 | 1.000 |
| 0.7 | 0.000 | 0.229 | 0.333 | 0.455 | 0.583 | 0.702 | 0.800 | 0.873 | 0.924 | 0.957 | 1.000 |
| 0.8 | 0.000 | 0.276 | 0.400 | 0.539 | 0.673 | 0.785 | 0.868 | 0.924 | 0.959 | 0.979 | 1.000 |
| 0.9 | 0.000 | 0.327 | 0.471 | 0.621 | 0.753 | 0.852 | 0.917 | 0.957 | 0.979 | 0.991 | 1.000 |
| 1.0 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

Fig. 8. The membership grade values for $k = 4.5$ and $l = 0.5$.

if the average of all confidence values is greater than l , then the confidence value of the comment is assumed to be the MAX confidence value present in the set.

- *Step 2:* If Step 1 does not hold true, then if there exists at least one element in the set produced by Φ less than a threshold value ρ_1 , and if the average of all confidence values is less than or equal to l , then the confidence value of the comment is assumed to be the MIN confidence value present in the set.
- *Step 3:* If neither Step 1 nor Step 2 holds true, then the confidence value of a comment is assumed to be equal to the average of confidences.

Before proceeding in the detailed explanation of the previously described algorithm the two new variables, ρ_0 and ρ_1 need further explanation, because they play an important part in the question answering system.

In order to assess the pertinence to a subject of interest of a database, and, more importantly, to distinguish which records should ultimately be kept for the development of algorithms and further investigations, it is required to establish thresholds. As mentioned previously (in the description of the system), the output of the fuzzy evaluator is the confidence vector that is the input of the confidence analyzer. Given that the confidence vector contains values between 0 and 1,

in order for the confidence analyzer to be able to exclude unacceptable records and answer questions, cut points that group the records into distinct sets, are required. If, for example, we assume that three regions are needed for the investigation denoting definite relevance, definite irrelevance to the subject of interest, and indecision, then two values are required. The first one, denoted by ρ_0 , determining the acceptable degree of relevance and used as the lower bound of its inclusion, and the second one, ρ_{sub1} , used as the upper bound of inclusion of irrelevance. For example, if we assume that $\rho_0 = 0.66$ and $\rho_1 = 0.33$, then records having, in the confidence vector, a value less than 0.33 are considered irrelevant to the intended application; records having values between 0.33 and 0.66 are considered not certain if they are either relevant or irrelevant; and records with confidence values greater than 0.66 are considered relevant to the application. This is only one possible scenario, and is by no means the only one. It is entirely possible that only one value is required that distinguishes association from disassociation, and in such a case, $\rho_0 = \rho_1$.

The algorithm presented previously that attributes a confidence value to a comment containing more than two words is based on the assumption that the relevant or irrelevant actions are associated with the overall tendency of the comment (indicated by the average) and the presence of at least one dominant action. To clarify the operation of the algorithm, consider the following example.

Let us assume that a database entry contains the following comments.

- 1) "I fixed a syntax problem in module XYZ."
- 2) "I fixed two problems and changed the syntax of a module statement."

Let us further assume that we would like to establish which comments report functional changes in the design of a computer system. Finally, assume that the following words are considered relevant words to functional errors and that they have been assigned the following confidence values:

| Word | Membership Grade |
|---------|------------------|
| FIX | 0.9 |
| SYNTAX | 0.2 |
| PROBLEM | 0.75 |
| CHANGE | 0.7 |

The comments (r_1, r_2) in the modified database would be as follows.

- 1) r_1 : "FIX SYNTAX PROBLEM"
- 2) r_2 : "FIX PROBLEM CHANGE SYNTAX"

Using $l = 0.5$ and $k = 7.0$ for (1), and $\rho_0 = 0.66$ and $\rho_1 = 0.33$ for the two thresholds, the algorithm would compute the following. For the first comment (r_1):

$$\begin{aligned} < \text{FIX SYNTAX} > &= 0.4282 \\ < \text{SYNTAX PROBLEM} > &= 0.3100 \end{aligned}$$

Thus, $\mu_A(r_1) = \text{MIN}(0.4282, 0.31) = 0.31$, because the average of all pairs is less than l . For the second comment, the algorithm would compute the following:

$$\begin{aligned} < \text{FIX PROBLEM} > &= 0.9893 \\ < \text{PROBLEM CHANGE} > &= 0.9476 \\ < \text{CHANGE SYNTAX} > &= 0.2747 \end{aligned}$$

$\mu_A(r_2) = \text{max}(0.9893, 0.9476, 0.2747) = 0.9893$, because the average of all pairs is less than l . The algorithm produces the correct answer for both comments.

In addition to the selection of records presented previously and their use to answer some questions regarding a database under consideration, the result analyzer can be used to produce other useful work, such as how many comments exceed a prespecified degree of confidence, and a confidence value for the entire database (i.e., the average of the confidence values) that can be computed as follows:

$$\mu_A(X) = \frac{\sum_{i=1}^n \mu_A(r_i)}{n} \quad (2)$$

where X represents the entire database and n is the number of records in the database.

III. EVALUATION OF THE QUESTION ANSWERING SYSTEM

The fuzzy question answering system has been developed as a means of evaluating the bookkeeping library, which was developed for the IBM 4381 computer systems, and which was suspected of containing comments that describe functional changes, and consequently was suspected of being pertinent to the subject of functional testing [16]. The main objective of the library databases was to maintain and control data files for microcode and logic design. Anytime the database was accessed, independently of the reason, a comment, of 40 characters maximum length, was added that presumably described the cause of the access. Obviously, a number of accesses were for routine reasons, such as adding newly developed code in the microcode library, aggregating logic functions for unit testing, and building a hardware and microcode machine model for manufacturing releases. A number of accesses, however, corresponded to functional changes of the logic and microcode, and such changes need to be distinguished by applying the proposed system.

In validating the proposed system and assessing its accuracy, we proceeded as follows. First, the list of unique words and the frequency of the number of words and comments were generated for both hardware and microcode databases. A detailed description of the results are in Fig. 9. Consequently, the relevant word tables were extracted from the list of unique words. Fig. 10 shows the frequency of words of the two modified databases. As suspected, the frequency of comments of three or more words was relatively low for both databases.

We constructed four experimental databases, two for the hardware and two for the microcode libraries, with comments selected at random and excluding entries that could be interpreted correctly by the tool because of their confidence being equal to 1 or 0, the reason being that we need to evaluate the performance of the tool, and establish its limits in a

| Hardware Database | | Microcode Database | |
|------------------------------|------------|------------------------------|------------|
| Unique words | 3770 | Unique words | 7274 |
| number of words in a comment | percentage | number of words in a comment | percentage |
| 0 | 1.49 | 0 | 0.01 |
| 1 | 1.05 | 1 | 6.75 |
| 2 | 7.86 | 2 | 14.27 |
| 3 | 6.81 | 3 | 15.59 |
| 4 | 11.55 | 4 | 17.28 |
| 5 | 11.14 | 5 | 14.40 |
| 6 | 12.29 | 6 | 14.24 |
| 7 | 15.89 | 7 | 10.06 |
| 8 | 13.92 | 8 | 4.91 |
| 9 | 11.06 | 9 | 1.90 |
| 10 | 5.73 | 10 | 0.30 |
| 11 | 0.90 | 11 | 0.27 |
| 12 | 0.30 | 12 | 0.01 |

Fig. 9. Words per comment frequency of the original databases.

worse case scenario. This occurs when there is a possibility of interpretation that will occur when the confidence is different than 0 or 1. The new databases contain the following.

- The first hardware database, DB1, with 3450 comments, contains 1400 comments with zero relevant words (40.5%), 1250 comments with one relevant word (36.3%), and 800 comments with two or more relevant words (23.2%).
- The first microcode database, DB2, with 5700 comments, contains 2450 comments with zero relevant words (43%), 1750 comments with one relevant word (31%), and 1500 comments with two or more relevant words (26%).
- The second hardware database, DB3, with 2000 comments, contains 500 comments with zero relevant words (25%), 960 comments with one relevant word (48%), and 540 comments with two or more relevant words (27%).
- The second microcode database, DB4, with 2500 comments, contains 425 comments with zero relevant words (17%), 1025 comments with one relevant word (41%), and 1050 comments with two or more relevant words (42%).

DB1 and DB2 were selected to contain a higher percentage of records containing zero relevant words, because we wanted to verify the accuracy of the tool when no information had been preserved. DB3 and DB4 were selected to reflect the percentages of the original databases. All the comments of the new databases (DB1-DB4) were manually evaluated, and one of the three confidence values, 1, 0.5, and 0, was assigned to every comment. We applied this breakdown to verify the

closeness of the answer to the individual comments between the manual evaluation and the tool. The three values indicate the following.

| | |
|-----|---|
| 1 | Denotes a definite functional change (definite relevance) |
| 0.5 | It may or may not describe a functional change (indecision) |
| 0 | Denotes a definite no functional change, or the comment makes no sense (definite irrelevance) |

Using the relevant word tables, interviews were conducted with a number of developers who had participated in the design of the IBM 4381 computer systems; 22 hardware developers and 28 microcode developers were interviewed. The developers were asked to attribute a value between 0 and 100 indicating their degree of belief that a certain word, when used in a comment, was describing a functional error, a bug, or a change. The answers were scaled with a maximum value equal to 1, and the interviews were analyzed by an FEV processor [10] that calculated the FEV for each word in the relevant word table.

| Hardware Database | | Microcode Database | |
|---------------------------------------|------------|---------------------------------------|------------|
| Relevant words 104 | | Relevant words 146 | |
| number of relevant words in a comment | percentage | number of relevant words in a comment | percentage |
| 0 | 25.53 | 0 | 16.90 |
| 1 | 48.34 | 1 | 41.18 |
| 2 | 21.24 | 2 | 28.98 |
| 3 | 4.34 | 3 | 10.34 |
| 4 | 0.54 | 4 | 2.33 |
| 5 | 0.01 | 5 | 0.22 |
| | | 6 | 0.05 |

Fig. 10. Relevant words per comment frequency of the modified databases.

Based on the FEV of each word, the confidence of each comment of the databases was computed by the Fuzzy Evaluator, which was invoked with parameters $l = 0.5, k = 4.5$. (Comparable results were obtained for k values between 1 and 15 in [17].) To be able to attribute confidence value for the comments with more than two relevant words, we divided the confidence interval into three regions. The first region, definite irrelevance, included confidence values between 0 and 0.33, with 0.33 being ρ_1 . The second region, indecision, included values 0.33 to 0.66, with 0.66 being ρ_0 . The third region, definite relevance, included values 0.66 to 1.0. The output of the Fuzzy Evaluator consisted of a confidence vector that was an array of membership numbers for each entry in the databases.

In our analysis of the experimental databases, we wanted to verify the following.

- 1) How far apart are the confidence values of the comments given by the tool and the manual evaluation?
- 2) How close will the proposed system be to the manual evaluation when it is used to extract all the comments that are reporting functional changes and to exclude all comments that represent routine database accesses?
- 3) What is the percentage of records that corresponded to functional changes versus the percentage of records that did not? We also wanted to evaluate the closeness of the percentages between the tool and the manual evaluation.

The answer to the first question can be found in Fig. 11, and the answers to the second and third questions are in Fig. 12. Referring to Fig. 11, the disagreement between the tool and the manual evaluation have been reported for confidence values greater than 0.25, 0.3, and 0.5. The results of disagreement suggest the closeness of the evaluation of the tool with respect to the manual evaluation. In particular, the third row suggests that very few of the comments have been evaluated to opposite values by the tool when compared to the manual evaluation.

The tool accuracy is measured by considering all comments that have been given a confidence greater than 0.66 as representing functional changes, and comments that are either

| | | DB1 | DB2 | DB3 | DB4 | DB5 |
|---|-----------------------------|---------|---------|---------|---------|---------|
| 1 | Disagreement more than 0.25 | 14.02 % | 20.26 % | 16.40 % | 25.76 % | 21.10 % |
| 2 | Disagreement more than 0.30 | 9.04 % | 11.98 % | 9.70 % | 13.76 % | 17.70 % |
| 3 | Disagreement more than 0.50 | 2.37 % | 2.35 % | 2.65 % | 3.48 % | 4.50 % |

Fig. 11. Disagreement between tool and manual evaluation.

considered irrelevant or whose relevancy cannot be decided upon are considered not to represent functional testing (i.e., comments with confidence values less than 0.66 are considered routine database accesses).

To evaluate the tool accuracy, it is necessary to first evaluate the tool error and consequently use such an error to represent the disagreement between the manual evaluation and the tool and produce the agreement between the two evaluations that constitutes the tool accuracy. The tool accuracy can be found in the first row of Fig. 12, which was compiled to include the following as tool errors.

- The tool is in error when it includes comments that are not considered functional changes by the manual evaluation.
- The tool is in error when it excludes comments that are considered by the manual evaluation to be functional changes.

The findings indicate that the agreement between the tool and the manual evaluation is as low as 94.20% and as high as 95.67% (for the four sample databases DB1–DB4). This indicates that the tool is very accurate in its decisions to separate the database entries is relevant and irrelevant to functional testing regions. Finally, it is of interest to identify whether a database is pertinent to the subject of interest by establishing the percentage of pertinent comments to the functional testing.

| | | DB1 | DB2 | DB3 | DB4 | DB5 | DB6 | DB7 |
|---|--|---------|---------|---------|---------|---------|---------|---------|
| 1 | Tool Accuracy | 95.48 % | 95.67 % | 94.85 % | 94.20 % | 90.37 % | 95.63 % | 96.10 % |
| 2 | Func. Changes Reported by Tool | 30.78 % | 34.28 % | 38.85 % | 50.48 % | 77.66 % | 22.47 % | 35.54 % |
| 3 | Func. Changes Reported by Manual Eval. | 30.55 % | 35.56 % | 38.60 % | 52.28 % | 75.41 % | 22.13 % | 34.63 % |
| 4 | Tool-Manual | 0.23 % | 1.28 % | 0.25 % | 1.80 % | 2.25 % | 0.33 % | 0.90 % |

Fig. 12. Tool vs. manual evaluation.

The second and third row of Fig. 12 report such percentages for the four experimental databases (DB1-DB4) for functional changes computed for the tool (second row) and the manual evaluation (third row). The fourth row reports the absolute value of the deviation between the tool and the manual evaluation. The findings indicate that the tool and the manual evaluation are very consistent among each other, indicating that the tool can be used to establish the relevance of a database to a prespecified subject of interest. The results indicate for DB1 and DB2, as expected, because of the number of comments containing comments with 0 relevant words, approximately one-third of the comments were related to functional errors, implying that those databases were mostly used for routine accesses. In DB4, approximately one-half of the accesses were routine accesses. At first glance, it appears that DB3 should also have the same characteristics as DB4; but it does not. The reason for the difference most likely lies in the fact that the comments with 0 confidence for DB3 are 25.53%; for DB4, they are 16.90%. An interesting observation is that the more relevant words that exist in a comment, the greater the percentage of relevant comments to functional testing in a database. The previous statement is based upon the percentages of functional changes reported by the tool and functional changes reported by manual evaluation for DB1, DB2, DB3, DB4, and DB5 (discussed later and comprising at least three words per comment).

The results of our analysis indicate that the tool evaluated the comments closely to the manual evaluation, and that the system performed satisfactorily in the excluding and including comments relevant to the functional changes, and in its decision of what portion of the database has or has not been used for functional changes. It can be observed that the two databases containing zero relevant words (DB1 and DB2) have been more accurate than the other two databases. This indicates that when there are no relevant words, most likely no relevant actions have occurred. Additionally, we wanted to verify the accuracy of the algorithm for three or more relevant words for the following reason. Given that the structure of the

sentences, as well as their meaning, is not taken into account by the tool, and also given the fact that the algorithm in DB4, which contains the most comments, with at least three words (13%), performed slightly worse, we suspected that as the relevant words increase in number in a comment, then the tool accuracy decreases. We determined that a further investigation was necessary to establish the limits of the tool.

We suspected that a comment with less than three relevant words indicates that in most cases, the single action principle is prevalent, and that comments containing more than two relevant words may possibly indicate two things.

- 1) The likelihood of relevant actions intermingled with irrelevant actions per single database access may be high.
- 2) While the order of the relevant words has been preserved, a number of irrelevant words have been deleted and such words may be necessary to distinguish the association of relevant words. A consequence of the previous discussion is that it may be difficult to attribute association of actions, and the conclusions of the tool may be inaccurate.

In order to evaluate the performance of the tool, we selected all comments, 488 in total, that contained more than two relevant words from DB2, and used the tool to evaluate the new database. The reason why DB2 was selected is that such a database was extracted from the microcode library and contained the most comments with more than two relevant words, and we considered 488 comments to constitute a reasonable sample. The results of the analysis can be found in Figs. 11 and 12 in the column denoted as DB5. The results indicate that the performance of the tool is acceptable, but is inferior to the results of the tool applied to the other databases, because the disagreement for more than 0.5 confidence is 4.50%, the accuracy (90.37%) is lower, and the deviation (2.25%) between the tool and the manual inspection has been increased. Although the comments containing more than two relevant words have obviously diminished the overall precision of the tool, the overall deterioration for DB1, DB2, DB3, and DB4, and the overall degradation was not of a great extent

because of the low frequency of such comments. In particular, DB4 constituted the worst-case scenario for the experimental databases (13% of comments with three or more words), where the tool accuracy has been above 94%, suggesting that the tool will produce acceptable results.

Given that the system has been developed for a particular subject of interest operating on bookkeeping libraries, it is of interest to investigate its accuracy for other libraries that presumably have been developed for the development of another system that could be considered to have similar characteristics. More importantly, it is of interest to investigate the accuracy of the system without any changes, i.e., to use the same relevant words and their confidence values developed for the 4381 computer system, by examining such library databases for functional changes. Obviously, in experimenting with another database, a number of relevant words, such as abbreviations that are used only in the development of the new system and misspelled words that are related to the subject of interest, may be excluded. However, most of the words should remain valid, because we are interested in the same process, and the confidence attributed from one design group should be similar enough to grant a fair evaluation.

In testing the accuracy of the tool in the circumstances described above, we used the 9370 microcode library subsystem, which is similar to the 4381 library system. The analysis was conducted on two databases, DB6 with 1500 randomly selected records, and DB7 with 1000 records, also selected randomly, with the constraint of having at least one relevant word. The results of our analysis can be found in Fig. 12. The results reported in Fig. 12 indicate that the tool accuracy was slightly better, and strongly suggest that the proposed system will perform satisfactorily even without changes, implying that the tool does not require additional development effort when used in other databases with characteristics similar to the one for which it was developed.

IV. CONCLUDING REMARKS

In the previous sections, we presented a question answering system based on fuzzy logic. The tool provides the capability to assess if a database contains information pertinent to a subject of interest, and to exclude the irrelevant portion of the database. Each comment of the database is evaluated via a fuzzy evaluator that attributes a fuzzy membership value indicating its relationship to the subject of interest. An assessment is provided for the database as a whole; a decision is given indicating whether such a database is pertinent to the subject of interest; and comments that are considered as irrelevant to the subject can be discarded. Our evaluations performed on four experimental databases lead us to the following conclusions.

- Very few comments have been evaluated by the tool with a big disagreement from the manual evaluation, indicating that the tool closely respected our perception of which comments corresponded to routine accesses and which corresponded to functional errors.
- The tool is very close to the manual evaluation regarding the percentages of the database corresponding to

functional changes and routine accesses. Consequently, it can be used to evaluate the pertinence of a database to prespecified subject of interest.

- The accuracy of the tool in excluding comments not pertinent to, and in including comments pertinent to, the functional testing is extremely high, suggesting that the tool can be used as a means of exclusion of database entries not pertinent to prespecified subject of interest.

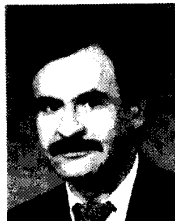
It should be noted that the reported tool accuracy possibly constitutes a worst-case scenario, because comments that will always be interpreted correctly, owing to their confidence being 0 or 1, have not been included in the accuracy evaluation. Additionally, our experiments with commented databases suggest that the proposed system can be used as a substitute for the natural language question answering system, because the fundamental assumptions for its development regarding comments in general hold true, and because comments with more than two relevant words are not sufficiently frequent to jeopardize its performance.

Finally, we establish that its applicability, assuming the same subject of interest, without modifications, is granted for databases comprising similar characteristics to the database for which the system had been previously applied, and we suggest that the system can be applied with minimum additional development effort to a variety of circumstances, provided that the fundamental assumptions for the development of the membership functions are respected in the new application.

REFERENCES

- [1] T. McNamara, "An application of cyclomatic complexity metrics to microcode," Tech. Rep. 01.A517, IBM Corp., Systems Product Div., Endicott, NY, USA, Mar. 1988.
- [2] G. Klir, "Is there more to uncertainty than some probability theorists might have us believe?" *Int. J. General Syst.*, vol. 15, pp. 347-378, 1989.
- [3] G. Klir and T. Folger, *Fuzzy Sets, Uncertainty, and Information*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [4] J. Corwin and J. Maggiore, Eds., "IBM 9370 Information System: Technology and development," Tech. Rep., IBM Corp., Systems Technology Div., Endicott, NY, USA, Dec. 1987.
- [5] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.
- [6] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic, 1980.
- [7] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. Reading, MA: Addison-Wesley, 1986, pp. 72-101.
- [8] R. Yager, S. Ovchinnikov, R. Tong, and H. Nguyen, Eds., *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh*. New York: Wiley, 1987.
- [9] A. Kaufmann and M. Gupta, *Introduction to Fuzzy Arithmetic: Theory and Applications*. New York: Van Nostrand Reinhold, 1985.
- [10] S. Vassiliadis, G. Triantafyllos, and W. Kobrosly, "CFEV: The Clustering Fuzzy Expected Value," IBM Tech. Rep. TR-01.C729, Endicott, NY, USA, Apr. 1993, p. 34.
- [11] A. Kandel, *Fuzzy Techniques in Pattern Recognition*. New York: Wiley, 1982, pp. 64-90.
- [12] M. Friedman, M. Schneider, and A. Kandel, "The use of Weighted Fuzzy Expected Value (WFEV) in fuzzy expert systems," *Fuzzy Sets and Syst.*, vol. 31, pp. 37-45, May 1989.
- [13] M. Smithson, *Fuzzy Set Analysis for Behavioral and Social Sciences*. New York: Springer-Verlag, 1986.
- [14] M. Kochen and A. Badre, "On the precision of adjectives which denote fuzzy sets," *J. Cybernetics*, vol. 4, no. 1, pp. 49-59, 1974.
- [15] P. Zysno, "Modeling membership functions," in B. Rieger, Ed., *Empirical Semantics*, vol. 1. Bochum, Germany: Studienverlag Brockmeyer, 1981, pp. 40-51.

- [16] S. Vassiliadis, G. Triantafyllos, and W. Kobrosly, "Establishing the relevancy of the bookkeeping libraries to the functional testing of computer implementations," IBM Tech. Rep. TR-01.C728, Endicott, NY, USA, Apr. 1993, p. 37.
- [17] G. Triantafyllos, W. Kobrosly, and S. Vassiliadis, "Algorithms for assessing the relevancy of a commented database to a subject of interest," IBM Tech. Rep. TR-01.B941, Endicott, NY, USA, July 1991, p. 43.



S. Vassiliadis (M'86-SM'92) received the Dr.Eng. degree in electronic engineering from the Politecnico di Milano, Milan, Italy, in 1978.

He is currently a faculty member in the Department of Electrical Engineering, Delft University of Technology, the Netherlands. He also has been a member of the faculty of Cornell University, Ithaca, NY, and the State University of New York, Binghamton, NY. During the 1992/93 academic year, he was a Visiting Professor in the School of Electrical Engineering, College of Engineering,

Cornell University. His previous working experience includes 10 years at IBM in the Advanced Workstations and Systems Laboratory in Austin, TX, the Mid-Hudson Valley Laboratory in Poughkeepsie, NY, and the Glendale Laboratory in Endicott, NY. At IBM, as a Chief Architect and Engineer, he provided guidance to a number of projects dealing with computer organizations and architectures, as well as the leadership in advanced research projects in a variety of computer related fields. He has been involved in the design and implementation of the IBM 9370 Model 60 computer system. His research interests include computer architecture, hardware design and functional testing of computer systems, parallel processors, computer arithmetic, EDFI for hardware implementations, neural networks, fuzzy logic and systems, and software engineering.

Dr. Vassiliadis is a member of the IEEE Computer Society. While at IBM, he received numerous awards, including 21 levels of the Publication Achievement Award, 14 levels of the Invention Achievement Award, and an Outstanding Innovation Award for Engineering/Scientific Hardware Design in 1989. Furthermore, in 1990, he was awarded the highest number of patents at IBM.



G. Triantafyllos received the Dipl. degree in electrical engineering from the National Technical Institute of Athens, Greece, in 1984, the B.S.E.E. degree from Fairleigh Dickinson University, Teaneck, NJ, USA, in 1987, the M.S. degree in computer engineering from Syracuse University, Syracuse, NY, USA, in 1989, and the Ph.D. in electrical engineering from the State University of New York, Binghamton, NY, USA, in 1993.

Since 1987, he has been employed at the Glendale Laboratories, IBM, Endicott, NY, and at Poughkeepsie Engineering Laboratory, IBM, Poughkeepsie, NY, where he has worked on assignments related to the automation of the functional testing, the reliability of computer systems, and the design of Open System Adapter for the IBM 3090 computer system. His research interests lie in the areas of computer architecture, hardware design, functional testing, software engineering, parallel processing, and fuzzy logic.

Since joining IBM, he has received five awards, published more than 30 papers and technical reports, and filed three patent applications.



W. Kobrosly received the B.A. degree in mathematics, the M.S. degree in computer science, and the Ph.D. degree in electrical engineering from the State University of New York (SUNY) at Binghamton, NY, USA, in 1976, 1978, and 1993, respectively. In addition, he received the M.S. degree in electrical engineering from Syracuse University, Syracuse, NY, USA, in 1985.

From 1978 to 1980, he worked at the Singer Corp., Link Flight Simulation Division, Binghamton, NY, USA, as a software engineer. From 1980 to the present, has been employed at IBM Corp., Endicott, NY, USA, and Austin, TX, USA. His current assignments at IBM Corp., Austin, TX, include test tools development, test automation, and test quantification. His research interests are in the general area of functional testing of computer systems.

He is a member of the IEEE Computer Society. Since joining IBM, he has received the First Patent Application Invention Achievement Award and the First and Second Publications Achievement Awards.