

GENERAL FUZZY AUTOMATA UNDER SEMANTICALLY SIMILAR ENVIRONMENT: AN APPLICATION TO ENHANCED SEARCH ENGINE PERFORMANCE

RANJEET KAUR^{1,2*}, ALKA TRIPATHI¹, §

ABSTRACT. Automata theory provides a systematic framework for modelling computational processes; however, many real-world systems require greater adaptability and flexibility. This paper presents a general fuzzy automata model ($GFA_{\mathbb{SE}}$) designed to address these challenges, particularly in environments where the set of states is not predetermined. The proposed model is capable of dynamically integrating semantically similar states based on contextual relevance, enhancing its robustness in handling real-world data variability. Algorithms are introduced to measure the similarity between states. This advancement extends the capabilities of existing General Fuzzy Automata (GFA) frameworks, allowing them to function effectively in uncertain or evolving environments. A practical application of this model is demonstrated in the domain of search engine performance, where the proposed approach improves the handling of queries by mapping them to semantically related states, thereby enhancing the relevance and accuracy of results.

Keywords: Fuzzy Automata, General Fuzzy Automata, Semantic Computing, Semantic Similarity, Search Engine.

AMS Subject Classification: 68Q45, 03D05, 68Q70.

1. INTRODUCTION

Theoretical computer science plays a significant role in many areas such as artificial intelligence, text processing, information retrieval, natural language processing, etc. [11, 39]. The significance of classical automata in theoretical computer science is widely acknowledged. Fuzzy sets introduced by [49] in 1965 have been extensively applied in automata

¹ Jaypee Institute of Information Technology- Department of Mathematics - Noida - 201016 - Uttar, Pradesh - India.

² ABES Engineering College Ghaziabad- 201009 - Uttar Pradesh - India.

e-mail: ranjeet.kaur@abes.ac.in, reetmaths@gmail.com;

ORCID: <https://orcid.org/0000-0002-1565-7051>.

e-mail: alka.choubey@gmail.com, alka.tripathi@mail.jiit.ac.in;

ORCID: <https://orcid.org/0000-0002-4442-3350>.

* Corresponding author

§ Manuscript received: June 06, 2025; accepted: October 07, 2025.

TWMS Journal of Applied and Engineering Mathematics, Vol.16, No.10; © Işık University, Department of Mathematics, 2026; all rights reserved.

theory. The concept of fuzzy automata was initially introduced by Santos [32] and thereafter formally proposed by Wee [43]. Fuzzy automata provide a framework for modelling systems that involve fuzzy or uncertain information, allowing for more flexible and adaptive behaviour compared to traditional automata [31, 33]. Subsequent research in fuzzy automata and language has been multi-directional, as documented in the literature. State minimization and reduction techniques for fuzzy automata have been studied in [5, 26, 35, 41] to reduce their complexity. Researchers have also focused on exploring the algebraic aspects of fuzzy automata and languages [18, 24, 25, 28, 34, 36, 37, 40]. Fuzzy automata have been used to identify similarities in approximate string matching [1, 4, 10] and for tokenization and pattern recognition in lexical analysis [20, 27]. Furthermore, the integration of fuzzy automata into emerging technologies, including neural networks [6] and fuzzy controllers [7], has been explored.

Semantic computing [38] holds significant importance in various domains due to its ability to enhance data interpretation, integration, and utilization. Semantic similarity and relatedness [22] are the two terms that play a significant role in applications of semantic computing. Semantic similarity emphasizes the likeness or resemblance between concepts based on their characteristics or features, while semantic relatedness evaluates the degree of association or connection between concepts, regardless of their specific attributes.

Traditional automata and fuzzy automata are defined on a predetermined environment, which is determined by the transition function and the set of inputs. It defines the behaviour and dynamics of the automaton as it progresses from one state to another based on the inputs. The concept of the environment becomes crucial in fuzzy automata, as it involves membership values of state transitions. To address the need for modelling applications with changes in state sets, one typically has to redefine automata to accommodate these changes. However, integration of semantic computing techniques and fuzzy automata can overcome these limitations by incorporating similar data based on similarity measures. Analyzing the environment of states in a fuzzy automaton helps in understanding the potential next states and their corresponding degrees of membership or possibility. This knowledge is valuable for decision-making, control strategies, or further analysis in applications involving imprecision or uncertainty, such as modeling human behaviour, linguistic systems, and expert systems.

In an effort to enhance the capabilities of automata theory, Ying [47] proposed a formal model of computing with words by considering inputs in the form of strings of an alphabet, rather than symbols. Cao et al. [3] discussed a generalized model of computing with words using retractions and generalized extension principles. Many researchers [2, 14, 15, 16, 30, 42] have formalized computing with words under various frameworks, including fuzzy automata, pushdown automata, stochastic automata, and probabilistic automata. Further, to enhance their capabilities, fuzzy automata are explored in the context of semantic computing to accommodate semantically similar input words. In 2019, Jiang [15] introduced the semantic computing of words with a view of automata theory and identified that finite automata and fuzzy finite automata can recognize two semantically related concepts based on their similarity. Huang et al. [12] laid the foundations for semantic computing based on probabilistic automata and discussed an application that incorporates similar data as input symbols, such as weather forecasting.

Wei et al. [44] proposed generalized fuzzy automata with semantic computing, where automata need to deal with states that are not in a predefined state set. Authors provided a method that can incorporate similar states into existing fuzzy automata instead of defining new fuzzy automata and discussed a real-time application based on weather forecast. Yadav and Tiwari et al. [46] introduced interval type-2 fuzzy finite rough automata for

computing with words and discussed an application in the medical diagnosis of patients. Recently, Kaur and Tripathi [19] proposed a hybrid model of GFA to accept inputs that are semantically similar to predefined words and discussed an application in transportation e-service. Authors [21] initially proposed a dynamic state model for GFA to incorporate similar states based on unobserved data, with an application discussed in air pollution forecasting.

1.1. Research Gap. The traditional automata and fuzzy automata have a limitation that the state set, once fixed, can not be changed. The existing automata become inadequate if there is a single change in the given state set. Although applications utilising GFA have been explored across several domains, such as natural language processing and artificial intelligence, the predetermined set of states may be insufficient to capture the dynamic characteristics of the data. In [44], changes in states are accommodated in existing fuzzy finite automata by considering only those states whose membership values of transition remain the same. The states with transitions of changed membership values are not considered for similarity.

1.2. Motivation. Investigating methods for fuzzy automata to learn from data could lead to more practical applications. Fuzzy automata that can adjust their behaviour based on a changing environment or user preferences could be beneficial. The integration of fuzzy automata with other computational models, such as semantic computing, might offer improved performance for handling uncertainty in real-world applications. The membership values of states in existing GFA, along with a similarity measure, can be used to define transitions for GFA_{SE} , enhancing its adaptability and accuracy. The importance of the membership value of states in GFA, particularly concerning semantically similar states, has a vast field of research.

1.3. Contribution. In the suggested study, GFA is presented as a computational model within a semantically similar environment to model and formalize real-time applications where predefining the state set is not feasible. In the proposed model, this issue is addressed by integrating semantic similarity to handle similar states. When an undefined state is encountered, rather than defining a new GFA, the proposed model allows the incorporation of the most similar known state in the existing GFA for a given problem. The principal contributions of this paper are summarised as follows:

- GFA is extended under a semantic similar environment to incorporate semantic similar states.
- Similarity between states is computed based on the membership values of states in existing GFA, which counts the transition value as well as the membership value of the state. Algorithms are proposed to compute similarity between states and to incorporate similar states into existing GFA.
- All the possible cases have been observed to compute similarity; the similar state is examined for source state, destination state or both.
- The idea of semantic computing is blended with fuzzy automata to support real-world applications where states are not fixed or can change at the run time.
- The proposed work is illustrated through a resource allocation application, where the model dynamically adjusts available resources in response to system overloading or malfunctioning, ensuring optimal resource utilization.

- Finally, a significant application is discussed in improving search engine performance for document retrieval. The model enables the retrieval of relevant documents by mapping user queries to contextually similar and related documents based on a similarity measure.

1.4. Organization of the paper. The paper is organized as follows: Section 2 provides an overview of basic definitions pertaining to semantic similarity, fuzzy automata and general fuzzy automata. Section 3 elaborates on the proposed method. Section 4 is aimed at enhancing search engine performance for document retrieval and ranking based on the proposed approach followed by the results and discussions in Section 5. Finally, Section 6 summarises conclusions and future work.

2. PRELIMINARIES

In this section, basic concepts associated with fuzzy automata, general fuzzy automata, and semantic similarity are recalled. The readers may refer [8, 9, 15, 17, 19, 21, 49] for details.

Definition 2.1. (*Fuzzy Set*) [23] Let $\mathfrak{X}$ be a universe of discourse. A fuzzy set $\mathfrak{A}$ defined on a set $\mathfrak{X}$, is a function that assigns each element of $\mathfrak{X}$ to a unique element of the interval $[0, 1]$, i.e. $\mathfrak{A} : \mathfrak{X} \rightarrow [0, 1]$.

For elements $\alpha, \beta \in [0, 1]$, $\alpha \vee \beta$ denotes $\max(\alpha, \beta)$ and $\alpha \wedge \beta$ denotes $\min(\alpha, \beta)$.

Some of the basic operations on fuzzy sets are:

- *scale product*: $(t \cdot \mathfrak{A}_1)(x) = \min(t, \mathfrak{A}_1(x))$;
- *product*: $(t \times \mathfrak{A}_1)(x) = t \times \mathfrak{A}_1(x)$;
- *union*: $(\mathfrak{A}_1 \cup \mathfrak{A}_2)(x) = \max(\mathfrak{A}_1(x), \mathfrak{A}_2(x))$;
- *intersection*: $(\mathfrak{A}_1 \cap \mathfrak{A}_2)(x) = \min(\mathfrak{A}_1(x), \mathfrak{A}_2(x))$;

where $\mathfrak{A}_1, \mathfrak{A}_2 \in \tilde{\mathcal{P}}(\mathfrak{X})$, $t \in [0, 1]$, $x \in \mathfrak{X}$.

Definition 2.2. [23] The fuzzy power set of $\mathfrak{X}$ denoted as $\tilde{\mathcal{P}}(\mathfrak{X})$, is the set of all fuzzy subsets $\mathfrak{A}$, which can be defined on the set $\mathfrak{X}$, $\tilde{\mathcal{P}}(\mathfrak{X}) = \{\mathfrak{A} \mid \mathfrak{X} \rightarrow [0, 1]\}$.

Definition 2.3. (*Height of Fuzzy Set*) [49]: The height of a fuzzy set $\mathfrak{A}$ is the largest membership value obtained by any element in that set.

Formally, $\text{height}(\mathfrak{A}) = \sup_{x \in \mathfrak{X}} (\mu_{\mathfrak{A}}(x))$.

Definition 2.4. (*Fuzzy Finite Automaton (FFA)*) [47] A FFA $\hat{\mathcal{M}} = (\hat{\mathcal{Q}}, \Sigma, \delta, \mathfrak{s}_0, \hat{\mathcal{F}})$ is a 5-tuple, where $\hat{\mathcal{Q}}$ is set of states, Σ is set of inputs, $\mathfrak{s}_0$ is initial state, $\hat{\mathcal{F}} \subseteq \hat{\mathcal{Q}}$ is fuzzy set of final states, and $\delta : \hat{\mathcal{Q}} \times \Sigma \rightarrow \tilde{\mathcal{P}}(\hat{\mathcal{Q}})$ is a mapping where $\tilde{\mathcal{P}}(\hat{\mathcal{Q}})$ denotes the set of all fuzzy subsets of $\hat{\mathcal{Q}}$.

Definition 2.5. (*Language accepted by FFA*) [47] The language $\mathcal{L}(\hat{\mathcal{M}})$ accepted by $\hat{\mathcal{M}}$ is a fuzzy subset of Σ^* and is defined by $\mathcal{L}(\hat{\mathcal{M}})(\theta) = \mathcal{L}(\hat{\mathcal{M}}, \theta)$ for all $\theta \in \Sigma^*$, where $\mathcal{L}(\hat{\mathcal{M}}, \theta) = \text{height}(\delta(\mathfrak{s}_0, \theta) \cap \hat{\mathcal{F}})$ is the degree to which θ is accepted by $\hat{\mathcal{M}}$.

Definition 2.6. (*General Fuzzy Automata (GFA)*) [8, 9, 17] A GFA $\mathcal{A} = \{\mathcal{Q}, \Sigma, \mathfrak{R}, \mathcal{Z}, \tilde{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2\}$ is an eight-tuple machine, where:

$\mathcal{Q}$ and Σ are finite non-fuzzy set of states and input symbols, $\mathfrak{R} \subseteq \tilde{\mathcal{P}}(\mathcal{Q})$ is the set of fuzzy start states; $\tilde{\mathcal{P}}(\mathcal{Q})$ is the set of all fuzzy subsets of $\mathcal{Q}$, $\mathcal{Z}$ is a finite non-fuzzy set of output symbols, $\tilde{\delta} : (\mathcal{Q} \times [0, 1]) \times \Sigma \times \mathcal{Q} \xrightarrow{\mathcal{F}_1(\mu, \delta)} [0, 1]$ is the augmented transition function (Aug_{tf}),

$\omega : \Omega \rightarrow \mathcal{Z}$ is the output function, $\mathcal{F}_1 : [0, 1] \times [0, 1] \rightarrow [0, 1]$ is membership assignment function, which is applied through $\tilde{\delta}$ to assign membership values to the active states, i.e. if there is a transition from state q_i to q_j with input $\mathbf{a}$, at time t then

$\mu^t(q_j) = \tilde{\delta}((q_i, \mu^{t-1}(q_i)), \mathbf{a}, q_j) = \mathcal{F}_1(\mu^{t-1}(q_i), \delta(q_i, \mathbf{a}, q_j))$ and $\mathcal{F}_2 : [0, 1]^* \rightarrow [0, 1]$ resolves the overlapping transitions of active states.

For $i \geq 0$, $\Omega_{act}(t_i)$ denotes set of all active states at time t_i . So, for $i = 0$, $\Omega_{act}(t_0) = \mathfrak{R}$ and $\Omega_{act}(t_i) = \{(q, \mu^{t_i}(q)) : \exists q' \in \Omega_{act}(t_{i-1}), \mathbf{a} \in \Sigma \mid \delta(q', \mathbf{a}, q) \in \Delta, \forall i \geq 1\}$; Δ represents the set of all transitions.

Definition 2.7. (Derivation of input string) [9] The derivation of an input string $\theta, \theta \in \Sigma^*$ denoted as $der_i(\theta); i \in N$, is a series of states that are traversed in sequential order, starting from the initial state to the final state such that each symbol of the string is processed.

Definition 2.8. (Active state set of input string) [9] The active state set of an input string θ , is the fuzzy set of all active states, after string θ has entered the GFA $\mathcal{A}$ and denoted as

$\Omega_{act}(\theta) = \{(q_{l_i}, \mu^{t=n_i-1}(q_{l_i})); q_{l_i} \text{ is active at time } t = n_i - 1 \text{ and } n_i \text{ is number of states in } der_i\}$.

Definition 2.9. (Semantic Similarity) [15, 19]: Semantic similarity measures the degree to which two concepts are alike in meaning. It is defined by the function $sim : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$ as

$$sim(u_1, u_2) = \begin{cases} 1, & \text{if } u_1 \equiv u_2 \\ s, & \text{if } u_1 \subseteq_R u_2, \text{ where } s \in (0, 1) \\ 0, & \text{if } u_1, u_2 \text{ are not related} \end{cases}.$$

Semantic similarity sim is usually represented as a score between 0 and 1, where a score of 0 means the concepts are not close at all, a score of 1 means they have identical meaning($\equiv$), and a value closer to 1 means concepts are more similar($\subseteq_R$).

Also, $sim(u_1, u_2) = sim(u_2, u_1)$.

The methods for defining the semantic similarity sim vary across different applications and are used to identify similar states to generalize the transition function of automata.

Definition 2.10. (Environment) [44] An automaton's environment is a collection of all the possible states that the automaton might encounter in its applications. Given an automaton $\mathcal{A}$ and its state set Ω , the environment where $\mathcal{A}$ works is defined as a set of states $\mathbb{S}$ such that $\Omega \subset \mathbb{S}$.

Definition 2.11. Consider two environments $\mathbb{S}$ and $\mathbb{S}'$. If for any $q \in \mathbb{S}$, there exists $q' \in \mathbb{S}'$ such that $sim(q, q') \in [0, 1]$, then $\mathbb{S}$ and $\mathbb{S}'$, are said to be semantically similar environment [21, 44] denoted as $\mathbb{S} \simeq \mathbb{S}'$ and the state q is said to be semantically similar to state q' , denoted as $q \sim q'$.

Remark 2.1. For environment $\mathbb{S}$ and automaton $\mathcal{A}$ with state set Ω , if $\Omega \in \mathbb{S}$, then we say automaton $\mathcal{A}$ can work in environment $\mathbb{S}$.

Example 2.1. Consider a GFA $\mathcal{A} = \{\Omega, \Sigma, \mathfrak{R}, \mathcal{Z}, \tilde{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2\}$ (Figure 1a) with environment $\mathbb{S}$. Then using Definition 2.10, the state set $\Omega = \{q_0, q_1, q_2\} \subseteq \mathbb{S}$ is called environment of states for GFA $\mathcal{A}$.

Now consider GFA $\mathcal{A}' = \{\Omega', \Sigma, \mathfrak{R}', \mathcal{Z}, \tilde{\delta}', \omega, \mathcal{F}_1, \mathcal{F}_2\}$ (Figure 1b) with environment $\mathbb{S}'$, where $\Omega' = \{q_0, q_2, p_1\} \subseteq \mathbb{S}'$. $\mathbb{S} \simeq \mathbb{S}'$, if $sim(q_1, p_1) \in [0, 1]$. In GFA $\mathcal{A}$ there is a transition from q_0 to q_1 with input $\mathbf{a}$ and a membership value of 0.8. However, in the GFA $\mathcal{A}'$ with environment $\mathbb{S}'$, there is a transition from q_0 to p_1 with input $\mathbf{a}$ but with a slightly different

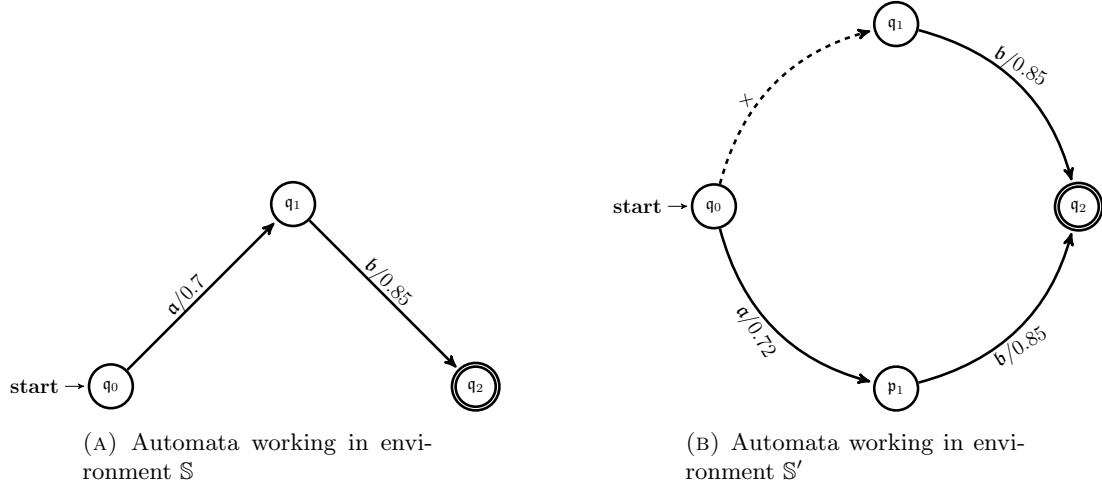


FIGURE 1. Similar Environment

membership value. The dotted line in the environment $\mathbb{S}'$ represents that though there is no direct transition from q_0 to q_1 but state q_1 still exists in the environment $\mathbb{S}'$. The similarity measure can be used to validate similar transitions, demonstrating that p_1 is similar to q_1 .

3. GENERAL FUZZY AUTOMATON UNDER SEMANTIC SIMILAR ENVIRONMENT

To model dynamic applications where the state set is not fixed, the idea of semantic computing in GFA is implemented, and a dynamic model of general fuzzy automaton under a semantically similar environment is proposed. The proposed model consists of an old state set as well as a new state set to accommodate similar states. An algorithm is proposed to compute the similarity between states. The membership values of states in a similar environment are determined by considering the similarity measure and membership value of states in the existing environment. Theorems have been proposed to prove the robustness and degree of acceptance of language accepted by GFA and $GFA_{\mathbb{SE}}$ under a semantically similar environment.

Definition 3.1. (General Fuzzy Automaton under Semantic Similar Environment)

A $GFA_{\mathbb{SE}}$ is a twelve-tuple machine $\tilde{\mathcal{A}} = (\mathcal{Q}_{old}, \mathcal{Q}_{new}, \mathcal{Q}_{all}, \Sigma, \mathcal{R}, \mathcal{R}', \mathcal{Z}, \tilde{\delta}, \hat{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2)$, where:

- (1) $\mathcal{Q}_{old}, \mathcal{Q}_{new}, \mathcal{Q}_{all}$ are finite non-fuzzy sets of states in existing environment, new environment and set of all possible states respectively, where $\mathcal{Q}_{old} \subseteq \mathcal{Q}_{all}$ and $\mathcal{Q}_{new} \subseteq \mathcal{Q}_{all}$,
- (2) $\mathcal{R} \subset \tilde{\mathcal{P}}(\mathcal{Q}_{old})$ and $\mathcal{R}' \subset \tilde{\mathcal{P}}(\mathcal{Q}_{all})$ are the set of fuzzy starting states in old and similar environments respectively,
- (3) $\Sigma = \{a_1, a_2, \dots, a_m\}$ is a finite non-fuzzy sets of input alphabets,
- (4) $\mathcal{Z}$ is a finite non-fuzzy set of output alphabets,
- (5) $\tilde{\delta} : (\mathcal{Q}_{old} \times [0, 1]) \times \Sigma \times \mathcal{Q}_{old} \xrightarrow{\mathcal{F}_1(\mu, \delta)} [0, 1]$ and $\hat{\delta} : (\mathcal{Q}_{all} \times [0, 1]) \times \Sigma \times \mathcal{Q}_{all} \xrightarrow{\mathcal{F}_1(\mu, \hat{\delta})} [0, 1]$ are the augmented transition functions in existing and similar environments respectively. The transition in new environment corresponding to transition $\delta(q, a, p) = \alpha$; $\alpha \in [0, 1]$, $p, q \in \mathcal{Q}_{old}$, $a \in \Sigma$ are defined considering four possible cases for

- $\beta \in [0, 1]$:
- CaseI:** For $p^* \sim p$; $p^* \in \mathfrak{Q}_{old}$, $\hat{\delta}(q, a, p^*) = \beta$
- CaseII:** For $p^{**} \sim p$;
 $p^{**} \in \mathfrak{Q}_{new}$, $\hat{\delta}(q, a, p^{**}) = \beta$
- CaseIII:** For $q^* \sim q$;
 $q^* \in \mathfrak{Q}_{old}$, $\hat{\delta}(q^*, a, p) = \beta$
- CaseIV:** For $q^{**} \sim q$;
 $q^{**} \in \mathfrak{Q}_{new}$, $\hat{\delta}(q^{**}, a, p) = \beta$
- (6) $\mathcal{F}_1 : [0, 1] \times [0, 1] \rightarrow [0, 1]$ the membership assignment function assigns membership values to active states. $\mathcal{F}_1(\mu, \hat{\delta})$ depends on membership value μ of prior state and the transition weight of $\hat{\delta} : \mathfrak{Q}_{all} \times \Sigma \times \mathfrak{Q}_{all} \rightarrow [0, 1]$. For transition, $\delta(q_i, a, q_j)$, $\mu^t(q_j) = \hat{\delta}((q_i, \mu^{t-1}(q_i)), a, q_j) = \mathcal{F}_1(\mu^{t-1}(q_i), \hat{\delta}(q_i, a, q_j))$ denotes the membership value of q_j at time t ,
- (7) $\mathcal{F}_2 : [0, 1]^* \rightarrow [0, 1]$ is called the multi-membership resolution function, which resolves the multi-membership active states and assigns a single membership value to active states.

Generally, the operations for $\mathcal{F}_1, \mathcal{F}_2$ are taken as minimum, maximum or average.

$\tilde{\mathfrak{Q}}_{act}(t_i)$ represents set of all active states at time $t_i, \forall i \geq 0$.

$\tilde{\mathfrak{Q}}_{act}(t_0) = \mathfrak{R}'$, $\tilde{\mathfrak{Q}}_{act}(t_i) = \{(q, \mu^{t_i}(q)) : \exists q' \in \tilde{\mathfrak{Q}}_{act}(t_{i-1}), \exists a \in \Sigma, \hat{\delta}(q', a, q) \in \Delta, \forall i \geq 1\}$. Δ denotes the set of all transitions of $GFA_{\mathbb{SE}}$.

Definition 3.2. (Language Accepted by $GFA_{\mathbb{SE}}$) A string of input symbols $\theta \in \Sigma^*$ is said to be acceptable by $GFA_{\mathbb{SE}}$, if starting from an initial state, at least one of the states in the active state set of θ is final.

θ is accepted by $\tilde{\mathcal{A}}$, $\iff \exists q_{l_i}; q_{l_i} \in \tilde{\mathfrak{Q}}_{act}(\theta) \cap \tilde{\mathfrak{Q}}_{final}$ in i^{th} derivation, where $\tilde{\mathfrak{Q}}_{final} \subseteq \mathfrak{Q}_{all}$ is the set of final states.

The final membership value of an acceptable string in $GFA_{\mathbb{SE}}$ $\tilde{\mathcal{A}}$ is calculated by acceptance calculation function $\mathcal{F}_3 : [0, 1]^* \rightarrow [0, 1]$ denoted as

$$\mu(\theta) = \mathcal{F}_3[\mu^{n_i-1}(q_{l_i})], \text{ and } \mathcal{L}(\tilde{\mathcal{A}})(\theta) = \{(\theta, \mu(\theta))\}_{q_{l_i} \in \tilde{\mathfrak{Q}}_{act}(\theta) \cap \tilde{\mathfrak{Q}}_{final}}$$

To establish the robustness and degree of acceptance of the language accepted by GFA and $GFA_{\mathbb{SE}}$ under a semantically similar environment, the following theorems are proposed.

Theorem 3.1. Suppose $\tilde{\mathcal{A}} = (\mathfrak{Q}_{old}, \mathfrak{Q}_{new}, \mathfrak{Q}_{all}, \mathfrak{R}, \mathfrak{R}', \Sigma, \mathcal{Z}, \mathcal{Z}', \tilde{\delta}, \hat{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2)$, is a $GFA_{\mathbb{SE}}$ that works in environment $\mathbb{S}$. Then $\tilde{\mathcal{A}}$ can work in semantic similar environment $\mathbb{S}'$.

Proof. If $\tilde{\mathcal{A}}$ works in environment $\mathbb{S}$, then $\mathfrak{Q}_{all} \subset \mathbb{S}$.

For any $q, p \in \mathfrak{Q}_{all}$, suppose there exists a transition $\hat{\delta}(q, a, p) = \alpha$; $\alpha \in (0, 1]$.

Let $q' \in \mathbb{S}'$ then, for semantic similar environment $\mathbb{S}'$ by definition 2.9 and definition 2.10 there exists a transition $\hat{\delta}(q', a, p) = \beta$; $\beta \in (0, 1]$ such that $sim(q, q') \in [0, 1]$.

Let $\mathfrak{Q}^* = \{q' \mid 0 \leq sim(q, q') \leq 1, q \in \mathfrak{Q}_{all}, q' \in \mathbb{S}'\}$ and

$$q^* = \{q' \mid sim(q, q') = \max_{\hat{q} \in \mathfrak{Q}^*} sim(q, \hat{q})\}.$$

Obviously, $\mathfrak{Q}^* \subset \mathbb{S}'$.

According to transition function of $GFA_{\mathbb{SE}}$ $\hat{\delta} : (\mathfrak{Q}_{all} \times [0, 1]) \times \Sigma \times \mathfrak{Q}_{all} \xrightarrow{\mathcal{F}_1(\mu, \hat{\delta})} [0, 1]$

Hence, $\tilde{\mathcal{A}}$ can work in $\mathfrak{Q}^*$, i.e. $\tilde{\mathcal{A}}$ can work in similar environment $\mathbb{S}'$. $\square$

Algorithm 1 is proposed to compute the similarity between similar states of the given GFA $\mathcal{A} = \{\mathfrak{Q}, \Sigma, \mathfrak{R}, \tilde{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2\}$ with n states.

Algorithm 1 To calculate similarity between states in old and new environment and membership values of states in new environment

Input: GFA $\mathcal{A} = \{\mathfrak{Q}, \Sigma, \mathfrak{R}, \mathcal{Z}, \tilde{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2\}$ with n states, set of states $\mathfrak{Q}$ and string $\theta = \theta_1\theta_2...\theta_l : \theta_i \in \Sigma$.

Output: Membership values of similar states.

Step-1 Consider the starting state set $\mathfrak{R} = \{(q_k, \mu^{t=0}(q_k) = 1)\}, 1 \leq k \leq n$

Step-2 For $1 \leq i \leq |\theta| = l$

Step-2.1 For input θ_i , find all the possible states active at time $t = i$; $\mathfrak{Q}_{act}(t = i)$.

Step-2.2 Calculate $\mu^{t=i}(\mathbf{p}) = \mathcal{F}_2[\mathcal{F}_1(\mu^{t=i-1}(q_j), \delta(q_j, \theta_i, \mathbf{p}))]; q_j \in \mathfrak{Q}_{act}(t = i-1)$;

$\mathbf{p} \in \mathfrak{Q}_{act}(t = i)$ i.e. the membership values of all active states using definition 2.8.

Step-3 Calculate membership value for language θ i.e.,

$\mu_{\mathcal{A}}(\theta) = \mathcal{F}_3[\mu^{t=l}(\mathbf{r}_m)]; \forall \mathbf{r}_m \in \mathfrak{Q}_{act}(t = l)$ and assign $\mathcal{L}_{\mathcal{A}}(\theta) = (\theta, \mu_{\mathcal{A}}(\theta))$.

Step-4 Construct $GFA_{SE} \tilde{\mathcal{A}} = (\mathfrak{Q}_{old}, \mathfrak{Q}_{new}, \mathfrak{Q}_{all}, \mathfrak{R}, \mathfrak{R}', \Sigma, \mathcal{Z}, \mathcal{Z}', \tilde{\delta}, \hat{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2)$, for changes in environment consider state set $\mathfrak{Q}_{new}$, $\mathfrak{Q}_{old} = \mathfrak{Q}$, set of all possible states as $\mathfrak{Q}_{all}$ and input threshold value ths_{val} .

Step-5 Consider starting state set in the new environment $\mathfrak{R}' = \{(q'_k, \mu^{t=0}(q'_k) = 1)\}, 1 \leq k \leq n$.

Step-6 For $1 \leq i \leq |\theta| = l$

Step-6.1 For input θ_i find all active states on the basis of transitions $\hat{\delta}$ i.e. $\tilde{\mathfrak{Q}}_{act}(t = i) \subseteq \mathfrak{Q}_{all}$.

Step-6.2 Calculate $\mu^{t=i}(q') = \mathcal{F}_2[\mathcal{F}_1(\mu^{t=i-1}(\mathbf{r}'_j), \hat{\delta}(\mathbf{r}'_j, \theta_i, q'))];$

$\mathbf{r}'_j \in \tilde{\mathfrak{Q}}_{act}(t = i-1), q' \in \tilde{\mathfrak{Q}}_{act}(t = i)$

Step-7 For $t = 0$ to $|\theta| = l$

Step-7.1 Calculate $|\mu^t(q') - \mu^t(q)| = err; q \in \mathfrak{Q}_{act}(t), q' \in \tilde{\mathfrak{Q}}_{act}(t)$

Step-7.2 If $err = 0, q = q'$, Assign $\mu^t_{\tilde{\mathcal{A}}}(q') = \mu^t(q)$.

Step-7.3 Else if $err < ths_{val}$, then $q' \sim q$

Step-7.3.1 Calculate $sim(q', q) = 1 - err$ and replace q' with q .

Step-7.3.2 Assign $\mu^t_{\tilde{\mathcal{A}}}(q') = sim(q', q) \times \mu^t(q)$.

Step-7.4 Otherwise q' and q are not similar.

Step-8 End

Theorem 3.2. Suppose $\mathcal{A} = \{\mathfrak{Q}, \Sigma, \mathfrak{R}, \mathcal{Z}, \tilde{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2\}$ is a general fuzzy automaton working in environment $\mathbb{S}$. $\tilde{\mathcal{A}} = (\mathfrak{Q}_{old}, \mathfrak{Q}_{new}, \mathfrak{Q}_{all}, \mathfrak{R}, \mathfrak{R}', \Sigma, \mathcal{Z}, \mathcal{Z}', \tilde{\delta}, \hat{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2)$, is a general fuzzy automaton under semantic similar environment $\mathbb{S}'$, i.e. $\mathbb{S} \simeq \mathbb{S}'$. Then for $\theta \in \Sigma^*$, $\mathcal{L}(\tilde{\mathcal{A}})(\theta) \subseteq \mathcal{L}(\mathcal{A})(\theta)$.

Proof. Let $\theta \in \Sigma^*$ and $\mathcal{L}(\mathcal{A})(\theta) = \{\theta, \mu(\theta)\}$; θ having i derivations, $der_i(\theta)$.

Hence, there exists q_{l_i} such that $q_{l_i} \in \mathfrak{Q}_{act}(\theta) \cap \tilde{\mathfrak{Q}}_{final}$.

$\implies q_{l_i}$ is active at time $t = n - 1$, where n is number of states in derivation der_i .

Hence, there exists transition $\tilde{\delta}((q_j, \mu^{t=n-2}(q_j)), \theta, q_{l_i}) = \mu^{t=n-1}(q_{l_i})$ where,

$q_j \in \tilde{\mathfrak{Q}}_{act}(t = n - 2); 1 \leq j < m, \mu^{t=n-1}(q_{l_i}) \in [0, 1]$.

Since $\tilde{\mathcal{A}}$ is a general fuzzy automaton under semantic similar environment $\mathbb{S}'$, therefore $\mathfrak{Q}_{old} = \mathfrak{Q}$ and for $q_{l_i} \in \mathfrak{Q}$ there exists $q'_{l_i} \in \mathfrak{Q}_{all} \subset \mathbb{S}'$ such that $sim(q_{l_i}, q'_{l_i}) = \beta_i; \beta_i \in [0, 1]$.

As $\theta \in \mathcal{L}(\tilde{\mathcal{A}})$, from algorithm 1, $\mu_{\tilde{\mathcal{A}}}^{t=n-1}(q'_{l_i}) = sim(q_{l_i}, q'_{l_i}) \times \mu^{t=n-1}(q_{l_i}) \leq \mu^{t=n-1}(q_{l_i})$
 $\mu_{\tilde{\mathcal{A}}}(\theta) = \mathcal{F}_3[\mu_{\tilde{\mathcal{A}}}^{n-1}(q'_{l_i})]$, and $\mathcal{L}(\tilde{\mathcal{A}})(\theta) = \{(\theta, \mu_{\tilde{\mathcal{A}}}(\theta))\}$
 $q'_{l_i} \in \tilde{\mathfrak{Q}}_{act}(\theta) \cap \tilde{\mathfrak{Q}}_{final}$

Hence, $\mathcal{L}(\tilde{\mathcal{A}}) \subseteq \mathcal{L}(\mathcal{A})$. □

To understand the concept of a dynamic environment, an application of resource allocation in a distributed cloud computing environment is modelled using GFA. Consider the scenario where one resource malfunctions, prompting the addition of new resources to replace the non-working resource. Alternatively, new resources might be introduced to the existing model to meet process requests. In this context, the same GFA will accommodate similar states, determined by the mem_vals of existing states, without defining a new automaton in response to resource changes. Consequently, the model can adapt to similar environmental shifts, ensuring uninterrupted operation instead of halting for gradual or predictable changes. However, if the changes require the addition of resources with entirely different characteristics, the adaptation may be less effective or even unsuitable.

Example 3.1. *The resource allocation process in a distributed cloud computing environment is modelled using traditional GFA. Consider GFA, $\mathcal{A} = (\mathfrak{Q}, \Sigma, \mathfrak{R}, \mathcal{Z}, \delta, \omega, \mathcal{F}_1, \mathcal{F}_2)$, where:*

- (1) *The state set $\mathfrak{Q} = \{q_0, q_1, q_2, q_3, q_4\}$ represents available servers, where*
 - q_0 : Initial load balancer (dispatcher).
 - q_1 : Server A (handles computationally light tasks).
 - q_2 : Server B (handles moderate tasks).
 - q_3 : Server C (handles resource-intensive tasks).
 - q_4 : Backup server (used for overflow or high-priority tasks).
- (2) $\Sigma = \{t_1, t_2, t_3\}$ *represents different tasks received by the system, where*
 - t_1 : Data pre-processing tasks.
 - t_2 : Machine learning model training tasks.
 - t_3 : Real-time data analytics tasks.
- (3) $\mathfrak{R} = \{(q_0, \mu^{t=0}(q_0) = 1)\}$ *represents the starting state, where all incoming requests are first sent to the load balancer. The initial membership value $\mu^{t=0}(q_0) = 1$ indicates that the load balancer is fully operational at the start.*

Transitions between states represent the time required to allocate and process tasks from one resource to another. The transitions between states (Figure 2) represent the time required to allocate and process tasks from one resource to another. A higher membership value in transition indicates that the resource requires less time to process the request. The transitions are defined as,

$$\begin{aligned} \delta(q_0, t_1, q_1) &= 0.9, & \delta(q_0, t_1, q_2) &= 0.8, & \delta(q_1, t_2, q_3) &= 0.6, & \delta(q_1, t_2, q_4) &= 0.82, \\ \delta(q_1, t_3, q_4) &= 0.7, & \delta(q_2, t_2, q_3) &= 0.6, & \delta(q_2, t_2, q_4) &= 0.9, & \delta(q_2, t_3, q_3) &= 0.8. \end{aligned}$$

To process the request(string) $\theta = t_1 t_2$ through different resources the possible operations according to GFA $\mathcal{A}$ using algorithm 1 are:

At time $t = 0$ active states are starting states $\mathfrak{Q}_{act}(t_0) = \{(q_0, \mu^{t=0}(q_0))\}$.

Since $\mathfrak{Q}_{act}(t_0) = \mathfrak{R}$ therefore, $\mu^{t=0}(q_0) = 1$

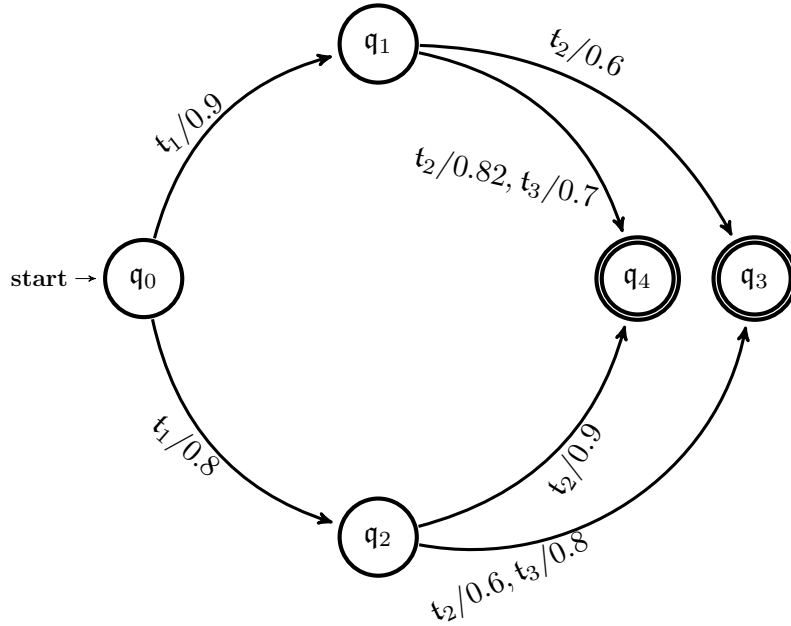


FIGURE 2. General Fuzzy Automaton for Resource Allocation

At time $t = 1$ on input t_1 active states are based on transitions from state q_0 .

As $q_0 \rightarrow q_1, q_2$, hence active states are $\{q_1, q_2\}$, therefore

$$\mu^{t=1}(q_1) = \mathcal{F}_1(\mu^{t=0}(q_0), \delta(q_0, t_1, q_1)) = \min(1, 0.9) = 0.9$$

$$\mu^{t=1}(q_2) = \mathcal{F}_1(\mu^{t=0}(q_0), \delta(q_0, t_1, q_2)) = \min(1, 0.8) = 0.8$$

Similarly, at time $t = 2$ for input t_2 , as $q_1 \rightarrow q_3, q_4$; $q_2 \rightarrow q_3, q_4$, hence active states are $\{q_3, q_4\}$, therefore

$$\begin{aligned} \mu^{t=2}(q_3) &= \mathcal{F}_2[\mathcal{F}_1(\mu^{t=1}(q_1), \delta(q_1, t_2, q_3)), \mathcal{F}_1(\mu^{t=1}(q_2), \delta(q_2, t_2, q_3))] \\ &= \max[\min(0.9, 0.6), \min(0.8, 0.6)] = 0.6 \end{aligned}$$

$$\begin{aligned} \mu^{t=2}(q_4) &= \mathcal{F}_2[\mathcal{F}_1(\mu^{t=1}(q_1), \delta(q_1, t_2, q_4)), \mathcal{F}_1(\mu^{t=1}(q_2), \delta(q_2, t_2, q_4))] \\ &= \max[\min(0.9, 0.82), \min(0.8, 0.9)] = 0.82 \end{aligned}$$

The membership values of states active at time $t = 0, 1, 2$ for input string $\theta = t_1 t_2$ are listed in Table 1.

TABLE 1. Membership values of states for resource allocation

time	0	1		2	
$Q_{act}(t)$	q_0	q_1	q_2	q_3	q_4
membership value	1	0.9	0.8	0.6	0.82

The membership value of language $t_1 t_2$ accepted by $\mathcal{A}$ is calculated using $\mathcal{F}_3$.

$$\begin{aligned} \mu_{\mathcal{A}}(t_1 t_2) &= \mathcal{F}_3[\mu^{t=2}(q_3), \mu^{t=2}(q_4)] \\ &= \max[0.6, 0.82] = 0.82 \end{aligned}$$

Hence, $\mathcal{L}(\mathcal{A})(t_1 t_2) = \{(t_1 t_2, 0.82)\}$.

Now consider the case where a resource is either removed due to malfunctioning or added due to an overload of processes. In response to this sudden change in demand, instead of abruptly terminating the allocation process, the model can explore alternative available resources within the existing GFA framework. This approach negates the need to create a new automaton. The same automata will accommodate similar states, where similarity is determined by the membership value of available states.

Consider $GFA_{SE} \tilde{A} = (\mathfrak{Q}_{old}, \mathfrak{Q}_{new}, \mathfrak{Q}_{all}, \Sigma, \mathfrak{R}, \mathfrak{R}', \mathcal{Z}, \tilde{\delta}, \tilde{\delta}', \omega, \mathcal{F}_1, \mathcal{F}_2)$ given in Figure 3, where $\mathfrak{Q}_{old} = \{q_0, q_1, q_2, q_3, q_4\}$, $\mathfrak{Q}_{new} = \{p_1, p_4\}$, with transitions

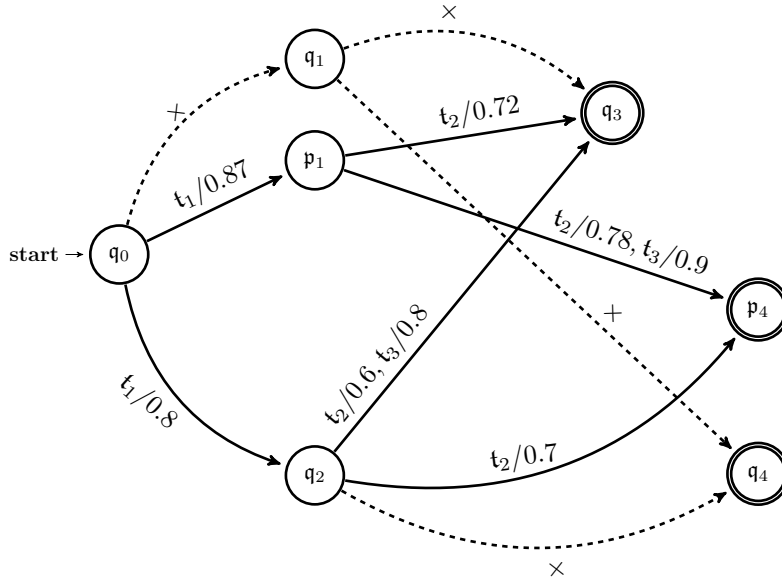


FIGURE 3. General Fuzzy Automaton under Semantic Similar Environment

$$\begin{aligned} \delta'(q_0, t_1, p_1) &= 0.87, & \delta'(q_0, t_1, q_2) &= 0.8, & \delta'(p_1, t_2, q_3) &= 0.72, & \delta'(p_1, t_2, p_4) &= 0.78, \\ \delta'(p_1, t_3, p_4) &= 0.9, & \delta'(q_2, t_2, q_3) &= 0.6, & \delta'(q_2, t_3, q_3) &= 0.8, & \delta'(q_2, t_2, p_4) &= 0.7 \end{aligned}$$

To process same request $t_1 t_2$ through different available resources, the possible derivations using Definition 2.7 for $GFA_{SE} \tilde{A}$ are:

$$der_1(t_1 t_2) = q_0 \rightarrow q_1 \rightarrow q_3,$$

$$der_2(t_1 t_2) = q_0 \rightarrow q_1 \rightarrow q_4,$$

$$der_3(t_1 t_2) = q_0 \rightarrow q_2 \rightarrow q_3,$$

$$der_4(t_1 t_2) = q_0 \rightarrow q_2 \rightarrow q_4$$

At time $t = 0$ active states using Definition 2.6 are starting states $\tilde{\mathfrak{Q}}_{act}(t_0) = \{(q_0, \mu^{t=0}(q_0))\}$. Since $\tilde{\mathfrak{Q}}_{act}(t_0) = \mathfrak{R}'$ therefore, $\mu^{t=0}(q_0) = 1$

At time $t = 1$ on input t_1 as $q_0 \rightarrow p_1, q_2$, hence active states are $\{p_1, q_2\}$, therefore

$$\mu^{t=1}(p_1) = \mathcal{F}_1(\mu^{t=0}(q_0), \delta'(q_0, t_1, p_1)) = \min(1, 0.87) = 0.87$$

$$\mu^{t=1}(q_2) = \mathcal{F}_1(\mu^{t=0}(q_0), \delta'(q_0, t_1, q_2)) = \min(1, 0.8) = 0.8$$

Similarly, at time $t = 2$ as $p_1 \rightarrow q_3, p_4$;

$q_2 \rightarrow q_3, p_4$, for input t_2 , hence active states are $\{q_3, p_4\}$, therefore

$$\mu^{t=2}(q_3) = \mathcal{F}_2[\mathcal{F}_1(\mu^{t=1}(p_1), \delta'(p_1, t_2, q_3)), \mathcal{F}_1(\mu^{t=1}(q_2), \delta'(q_2, t_2, q_3))]$$

$$= \max[\min(0.87, 0.72), \min(0.8, 0.6)] = \max[0.72, 0.6] = 0.72$$

$$\begin{aligned} \mu^{t=2}(\mathbf{p}_4) &= \mathcal{F}_2[\mathcal{F}_1(\mu^{t=1}(\mathbf{p}_1), \delta'(\mathbf{p}_1, \mathbf{t}_2, \mathbf{p}_4)), \mathcal{F}_1(\mu^{t=1}(\mathbf{q}_2), \delta'(\mathbf{q}_2, \mathbf{t}_2, \mathbf{p}_4))] \\ &= \max[\min(0.87, 0.78), \min(0.8, 0.7)] = \max[0.78, 0.7] = 0.78 \end{aligned}$$

The membership values of active states using Definition 2.8 are calculated and listed in Table 2.

TABLE 2. Membership values of states of $GFA_{\mathbb{SE}}$ for resource allocation

time	0		1		2
$Q_{act}(t)$	$\mathbf{q}_0$	$\mathbf{p}_1$	$\mathbf{q}_2$	$\mathbf{q}_3$	$\mathbf{p}_4$
membership value	1	0.87	0.8	0.72	0.78

Considering the threshold value 0.05 for tolerance of the minimum possible error to compute similarity between states using algorithm 1. For input alphabet $\mathbf{t}_1$ active states at time $t = 1$ are $\mathbf{q}_2, \mathbf{p}_1$.

$$\begin{aligned} \mu^{t=1}(\mathbf{p}_1) &= 0.87, \\ \text{Since, } |\mu^{t=1}(\mathbf{q}_1) - \mu^{t=1}(\mathbf{p}_1)| &= 0.03 < 0.05, \text{ and} \\ |\mu^{t=1}(\mathbf{q}_2) - \mu^{t=1}(\mathbf{p}_1)| &= 0.07 > 0.05, \\ \text{Therefore, } \mathbf{p}_1 &\sim \mathbf{q}_1. \end{aligned}$$

$$\begin{aligned} \text{sim}(\mathbf{p}_1, \mathbf{q}_1) &= 1 - 0.03 = 0.97 \\ \text{Hence, } \mu_{\tilde{\mathcal{A}}}^{t=1}(\mathbf{p}_1) &= 0.8439. \end{aligned}$$

Similarly calculating membership values for active states at time $t = 2$ for input $\mathbf{t}_2$,

$$\begin{aligned} \mu^{t=2}(\mathbf{p}_4) &= 0.78 \\ \therefore |\mu^{t=2}(\mathbf{q}_4) - \mu^{t=2}(\mathbf{p}_4)| &= 0.04 \leq 0.05, \\ \therefore \mathbf{p}_4 &\sim \mathbf{q}_4. \\ \text{Therefore, } \text{sim}(\mathbf{p}_4, \mathbf{q}_4) &= 1 - 0.04 = 0.96, \text{ hence } \mu_{\tilde{\mathcal{A}}}^{t=2}(\mathbf{p}_4) = 0.7488 \end{aligned}$$

The final membership values of states in $GFA_{\mathbb{SE}} \tilde{\mathcal{A}}$ are listed in Table 3. The mem-

TABLE 3. Final membership values of states of $GFA_{\mathbb{SE}}$ for resource allocation

time	0		1		2
$Q_{act}(t)$	$\mathbf{q}_0$	$\mathbf{p}_1$	$\mathbf{q}_2$	$\mathbf{q}_3$	$\mathbf{p}_4$
membership value ($\mu_{\tilde{\mathcal{A}}}^{t=n}$)	1	0.8439	0.8	0.72	0.7488

bership value for string or request $\mathbf{t}_1\mathbf{t}_2$ accepted by $GFA_{\mathbb{SE}} \tilde{\mathcal{A}}$ is calculated using $\mathcal{F}_3$.

$$\mu_{\tilde{\mathcal{A}}}(\mathbf{t}_1\mathbf{t}_2) = \mathcal{F}_3[\mu_{\tilde{\mathcal{A}}}^{t=2}(\mathbf{q}_3), \mu_{\tilde{\mathcal{A}}}^{t=2}(\mathbf{p}_4)] = \max[0.72, 0.7488] = 0.7488$$

Hence, $\mathcal{L}(\tilde{\mathcal{A}})(\mathbf{t}_1\mathbf{t}_2) = \{(\mathbf{t}_1\mathbf{t}_2, 0.7488)\}$. Using Theorem 3.2, $\mathcal{L}(\tilde{\mathcal{A}}) \subset \mathcal{L}(\mathcal{A})$, thus similar states are merged on the basis of highest similarity and membership value of the same language is contained in the language accepted by the given GFA.

4. APPLICATION

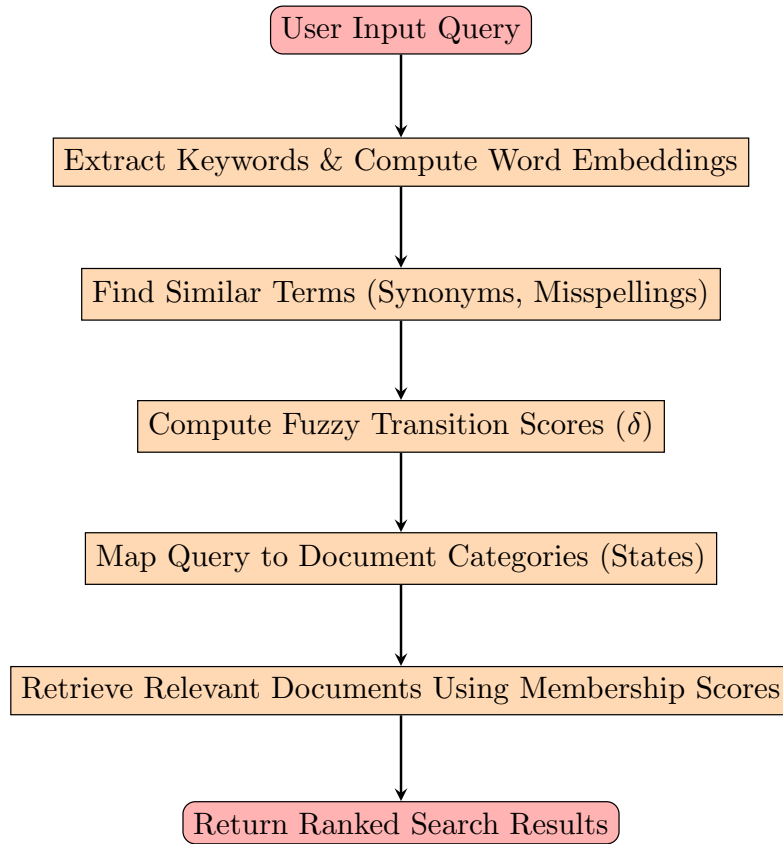
Search engines play a crucial role in retrieving relevant information from vast repositories of data. Traditional keyword-based search mechanisms often struggle with variations in user queries, such as synonyms, typographical errors, and semantically similar terms. These systems usually rely on exact keyword matching or simple statistical models, which may fail to capture the contextual similarity between different topics. By modelling topics as states in a fuzzy automaton, where transitions are defined based on similarity measures, semantically or contextually related topics can be activated even if they don't exactly match the query. This enables the system to retrieve more relevant documents that align with the intended meaning of the user's query, thereby overcoming the rigid nature of traditional systems.

This limitation can result in inaccurate or incomplete search results, negatively impacting the overall user experience. Traditional search engines typically use exact keyword matching or statistical models to retrieve results. However, users frequently submit queries containing misspellings, synonyms, or semantically related terms. This can lead to suboptimal results if the system does not intelligently map input words to relevant indexed documents. The proposed approach integrates GFA with SC to accommodate semantically similar input queries. The dynamic model of GFA and SC enhances document retrieval by mapping input terms into fuzzy transitions based on their semantic similarity, enabling flexible retrieval without rigid keyword dependencies. The proposed method improves search accuracy by accommodating word variations and dynamically mapping query terms to semantically relevant indexed documents.

Consider the GFA $\mathcal{A} = (\mathfrak{Q}, \Sigma, \mathfrak{R}, \mathcal{Z}, \tilde{\delta}, \omega, \mathcal{F}_1, \mathcal{F}_2)$. In the proposed model, when a user enters a query, contextually similar and related topics are identified and treated as states, which are then mapped to relevant documents. This modelling approach, based on fuzzy automata, enhances the efficiency of a search engine by incorporating both linguistic uncertainty and semantic similarity. In the proposed approach, user queries are represented as Σ , while states are categorised into topics and documents, with the query serving as the initial state. When a user enters a query, the initial state is activated at $t = 0$ with membership value 1. The working methodology is explained and illustrated using a flow-chart.

Working Methodology

- (1) Define $\Sigma = \{\mathcal{V}_i\}$ as the user query.
- (2) Subdivide the state set $\mathfrak{Q}$ into three categories:
 - Initial state Q_0 (representing the query input).
 - Intermediate states (representing topics or categories).
 - Final states (representing retrievable documents).
- (3) Each intermediate state corresponds to a topic, with associated keywords stored in a dictionary.
- (4) Documents are stored in a dictionary and retrieved based on their relevance to a category, determined by the maximum similarity score.



- (5) Each query is embedded into a vector space using the embedding function $\phi : \mathcal{V} \rightarrow R^d$, which maps each word to a high-dimensional vector representation (e.g., using Word2Vec or GloVe).
- (6) The cosine similarity [45] between each word $\mathbf{a} \in \mathcal{V}_i$ and indexed terms or keywords $\mathbf{b}$ in topic states (defined in Step 3) is computed as:

$$\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}$$
- (7) Based on the highest similarity values, fuzzy transitions are assigned to map the query to relevant states.
- (8) Document retrieval is done using Latent Dirichlet Allocation (LDA) [13], where similarity scores are computed based on the most frequent words relevant to each document. Transitions from intermediate topic states to document states are determined using these scores. The documents are sorted in descending order, ranking them from the most to the least relevant. This ranking ensures that users receive results that best match their intent, even if the query contains variations in wording or synonyms.

To validate the accuracy and effectiveness of the approach, the method is executed over 25 queries, including those with typo_errs and semantically related text. The results are analyzed, and document retrieval is done with the help of LDA algorithm [13] using built-in functions from Python libraries. Table 4 presents some of the similarity scores between various queries and five ranked documents. For each query, the document with the highest similarity score indicates the most relevant document retrieved for that specific query.

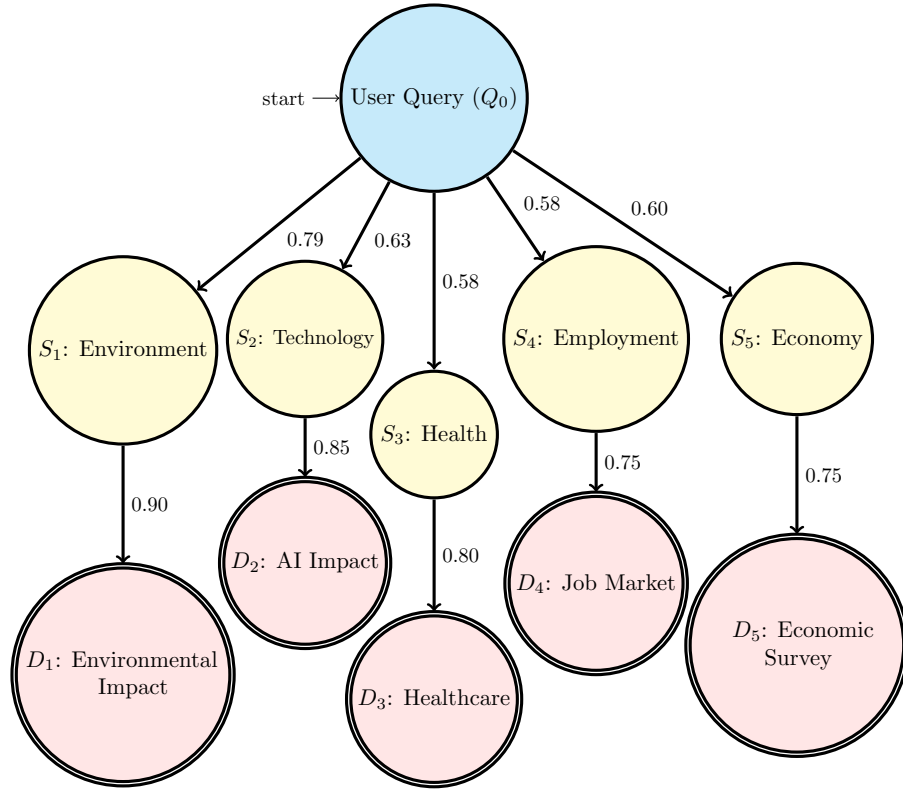
FIGURE 4. User query categorization using GFA_{SE}

TABLE 4. Query similarity scores for ranked document retrieval

Query	D_1	D_2	D_3	D_4	D_5
Solar energy	0.8451	0.3123	0.2145	0.1984	0.6432
Machine learning techniques	0.2312	0.9234	0.1145	0.3214	0.2845
Mental health awareness	0.1874	0.2398	0.8521	0.2687	0.1932
Data science jobs	0.1213	0.3412	0.1874	0.9156	0.2764
Financial investments	0.1984	0.2546	0.1321	0.1934	0.9231
Carbon footprint reduction	0.8912	0.2987	0.2012	0.1874	0.4872
AI in healthcare	0.2143	0.8123	0.7456	0.3128	0.2879

5. RESULTS & DISCUSSION

The methodology is implemented using Python with the GloVe model [29] for word embeddings. The states Environment (S_1), Technology (S_2), Health (S_3), Employment (S_4), and Economy (S_5) are considered as topics and documents Environmental Impact (D_1), AI Impact (D_2), Healthcare (D_3), Job Market (D_4), and Economic Survey (D_5) are mapped accordingly (Figure 4). When the query "solar energy" is processed, it is first converted to lowercase and filtered based on the GloVe vocabulary. If both words exist, they are transformed using the LDA dictionary. The query is then mapped to a topic distribution using the trained LDA model [13]. Next, the cosine similarity [45] between the query's topic vector and each document's topic vector is computed. The transitions in GFA_{SE} are determined based on the similarity score. Since "solar energy" is closely

related to the environmental domain, document D_1 (Environmental Impact) is ranked first, followed by D_5 .

6. CONCLUSION

The present work blends semantic computing with fuzzy automata theory, and a dynamic state model of general fuzzy automata is introduced for real-time applications where pre-defining state sets is impractical. An algorithm is proposed to accommodate similar states within semantically similar environments by computing their similarity and determining membership values. To validate, an application of resource allocation in the cloud computing environment is discussed that can incorporate available resources in response to system overloading or malfunctioning, ensuring optimal resource utilisation. Further, an application is proposed to enhance the performance of search engines. Instead of relying solely on exact state matches, the model maps user queries to semantically similar states. This dynamic state assignment allows for a more flexible and context-aware interpretation of search terms. To ensure fast and accurate document retrieval, the LDA algorithm is employed for effective document ranking based on topic similarity. Experimental results demonstrate the efficiency and accuracy of this approach, highlighting its potential to significantly improve the relevance and responsiveness of search engine results.

In the future, there is potential to explore the proposed model for a broader range of dynamic applications. For computing similarity, other measures may be explored, and by incorporating semantic similarity based on both input and states, the model can be further enhanced to improve its robustness. This approach would allow the model to better interpret and analyze data.

REFERENCES

- [1] Astrain, J. J., Garitagoitia, J. R., Gonzalez De Mendivil, J., Villadangos, J., Fariña, F., (2004), Approximate string matching using deformed fuzzy automata: A learning experience, *Fuzzy Optimization and Decision Making*, 3, 141–155.
- [2] Cao, Y., Xia, L., Ying, M., (2013), Probabilistic automata for computing with words, *Journal of Computer and System Sciences*, 79(1), 152–172.
- [3] Cao, Y., Ying, M., Chen, G., (2007), Retraction and generalized extension of computing with words, *IEEE Transactions on Fuzzy Systems*, 15(6), 1238–1250.
- [4] Choubey, A., Ravi, K., (2013), Minimization of deterministic finite automata with vague (final) states and intuitionistic fuzzy (final) states, *Iranian Journal of Fuzzy Systems*, 10(1), 75–88.
- [5] Ćirić, M., Stamenković, A., Ignjatović, J., Petković, T., (2010), Fuzzy relation equations and reduction of fuzzy automata, *Journal of Computer and System Sciences*, 76(7), 609–633.
- [6] Doostfateme, M., Kremer, S. C., (2004), Representing generalized fuzzy automata in recurrent neural networks, *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No. 04CH37513)*, 4, 1901–1906.
- [7] Doostfateme, M., Kremer, S. C., (2005), Developing a new fuzzy controller, *NAFIPS 2005 - Annual Meeting of the North American Fuzzy Information Processing Society*, 187–192.
- [8] Doostfateme, M., Kremer, S. C., (2005), New directions in fuzzy automata, *International Journal of Approximate Reasoning*, 38(2), 175–214.
- [9] Doostfateme, M., Kremer, S. C., (2006), General fuzzy automata, new efficient acceptors for fuzzy languages, *IEEE International Conference on Fuzzy Systems*, 2097–2103.
- [10] Garitagoitia, J. R., Gonzalez De Mendivil, J. R., Echanobe, J., Astrain, J. J., Fariña, F., (2003), Deformed fuzzy automata for correcting imperfect strings of fuzzy symbols, *IEEE Transactions on Fuzzy Systems*, 11(3), 299–310.
- [11] Hopcroft, J. E., Motwani, R., Ullman, J. D., (2001), *Introduction to Automata Theory, Languages, and Computation*, 2nd edition, ACM Press, New York, NY, USA.
- [12] Huang, G., Wasti, S. H., Wei, L., Jiang, Y., (2019), A formal model for semantic computing based on generalized probabilistic automata, *Entropy*, 21(9), 1–14.

- [13] Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., Zhao, L., (2019), Latent dirichlet allocation (LDA) and topic modeling: models, applications, a survey, *Multimedia Tools and Applications*, 78, 15169–15211.
- [14] Jiang, Y., (2018), A general type-2 fuzzy model for computing with words, *International Journal of Intelligent Systems*, 33(4), 713–758.
- [15] Jiang, Y., (2019), A formal model of semantic computing, *Soft Computing*, 23(14), 5411–5429.
- [16] Jiang, Y., Tang, Y., (2014), An interval type-2 fuzzy model of computing with words, *Information Sciences*, 281, 418–442.
- [17] Kaur, R., Tripathi, A., (2021), State minimization of general finite fuzzy automata, *International Journal of Mathematical, Engineering and Management Sciences*, 6(6), 1709–1718.
- [18] Kaur, R., Tripathi, A., (2022), General fuzzy multiset grammar and language, *Mathematics in Engineering, Science & Aerospace (MESA)*, 13(2), 1–10.
- [19] Kaur, R., Tripathi, A., (2024), Hybrid model of general fuzzy automata and semantic computing: An application to transportation e-service, *Soft Computing*, 28, 10519–10530.
- [20] Kaur, R., Tripathi, A., (2024), Statistical analysis of lexemes generated in ‘C’ programming using fuzzy automation, *Journal of Intelligent & Fuzzy Systems*, 46(1), 2537–2546.
- [21] Kaur, R., Tripathi, A., (2025), Dynamic state general fuzzy automata under semantically similar environment: An application-based study on air pollution for smart cities 4.0, *New Mathematics and Natural Computation*, 1–24.
- [22] Kim, J., Ostrowski, D. A., Yamaguchi, H., Sheu, P. C.-Y., (2013), Semantic computing and business intelligence, *International Journal of Semantic Computing*, 7(1), 87–117.
- [23] Klir, G. J., Yuan, B., (1994), *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice-Hall, USA.
- [24] Lee, E. T., Zadeh, L. A., (1969), Note on fuzzy languages, *Information Sciences*, 1(4), 421–434.
- [25] Li, Z., Li, P., Li, Y., (2006), The relationships among several types of fuzzy automata, *Information Sciences*, 176(15), 2208–2226.
- [26] Malik, D. S., (2002), *Fuzzy Automata and Languages: Theory and Applications*, Chapman and Hall/CRC.
- [27] Mateescu, A., Salomaa, A., Salomaa, K., Yu, S., (1995), Lexical analysis with a simple finite-fuzzy-automaton model, *Journal of Universal Computer Science*, 1(5), 292–311.
- [28] Mizumoto, M., Toyoda, J., Tanaka, K., (1969), Some considerations on fuzzy automata, *Journal of Computer and System Sciences*, 3(4), 409–422.
- [29] Pennington, J., Socher, R., Manning, C. D., (2014), GloVe: Global vectors for word representation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.
- [30] Qiu, D., Wang, H., (2005), A probabilistic model of computing with words, *Journal of Computer and System Sciences*, 70(2), 176–200.
- [31] Santos, E., (1976), Fuzzy automata and languages, *Information Sciences*, 10(3), 193–197.
- [32] Santos, E. S., (1968), Maximin automata, *Information and Control*, 13(4), 363–377.
- [33] Santos, E. S., (1975), Max-product grammars and languages, *Information Sciences*, 9(1), 1–23.
- [34] Shamsizadeh, M., Movahednia, E., De la Sen, M., (2023), Isomorphism between two vector general fuzzy automata, *Informatica*, 34(3), 617–633.
- [35] Shamsizadeh, M., Zahedi, M., (2015), Minimal intuitionistic general l-fuzzy automata, *Italian Journal of Pure and Applied Mathematics*, 35, 155–186.
- [36] Shamsizadeh, M., Zahedi, M. M., (2016), Intuitionistic general fuzzy automata, *Soft Computing*, 20, 3505–3519.
- [37] Shamsizadeh, M., Zahedi, M. M., (2019), Bisimulation of type 2 for bl-general fuzzy automata, *Soft Computing*, 23(20), 9843–9852.
- [38] Sheu, P. C.-Y., Yu, H., Ramamoorthy, C., Joshi, A. K., Zadeh, L. A., (2011), *Semantic Computing*, John Wiley & Sons.
- [39] Sipser, M., (2013), *Introduction to the Theory of Computation*, 3rd edn, Course Technology, Boston, MA.
- [40] Tiwari, S., Gautam, V., Dubey, M., (2016), On fuzzy multiset automata, *Journal of Applied Mathematics and Computing*, 51, 643–657.
- [41] Tripathi, A., Kaur, R., (2019), A review of state minimization and state reduction techniques in fuzzy automata, *AIP Conference Proceedings*, 2061(1), 1–7.
- [42] Wang, H., Qiu, D., (2003), Computing with words via Turing machines: A formal approach, *IEEE Transactions on Fuzzy Systems*, 11(6), 742–753.

- [43] Wee, W. G., Fu, K.-S., (1969), A formulation of fuzzy automata and its application as a model of learning systems, *IEEE Transactions on Systems Science and Cybernetics*, 5(3), 215–223.
- [44] Wei, L., Huang, G., Wasti, S. H., Hussain, M. J., Jiang, Y., (2021), Generalized fuzzy automata with semantic computing, *Soft Computing*, 25, 5775–5789.
- [45] Xia, P., Zhang, L., Li, F., (2015), Learning similarity with cosine similarity ensemble, *Information Sciences*, 307, 39–52.
- [46] Yadav, S., Tiwari, S., Kumari, M., Yadav, V. K., (2022), An interval type-2 fuzzy model of computing with words via interval type-2 fuzzy finite rough automata with application in COVID-19 deduction, *New Mathematics and Natural Computation*, 18(1), 61–101.
- [47] Ying, M., (2002), A formal model of computing with words, *IEEE Transactions on Fuzzy Systems*, 10(5), 640–652.
- [48] Garg, H., Ullah, K., Akram, M., Ali, Z., (2024), A Method To Multi-Attribute Group Decision Making Technique Based On T-Spherical Fuzzy Information Using Bonferroni Mean Operators, *TWMS J. Pure Appl. Math.*, 15(1), pp. 42-64.
- [49] Zadeh, L. A., (1965), Fuzzy sets, *Information and Control*, 8(3), 338–353.



Prof. Alka Tripathi is a distinguished academician and researcher, currently appointed as the Dean Academics UG at the Wish Town Campus, Jaypee Institute of Information Technology (JIIT). Prior to this, she served as Professor and Head of the Department of Mathematics at the JIIT, Noida, India. She earned her Ph.D. in Mathematics from Banaras Hindu University in 1995. With over 28 years of teaching and research experience, Prof. Tripathi has made significant contributions to both academia and research. Her primary research interests span Categorical Fuzzy Topology, Fuzzy Automata, Rough Sets. She has an impressive publication record, including numerous research papers in reputed international journals. Prof. Tripathi has guided six Ph.D. theses and is currently supervising three more.



Dr. Ranjeet Kaur is an accomplished academician and researcher in the field of mathematics, currently serving as Assistant Professor at ABES Engineering College Ghaziabad, India. She earned her Ph.D. in Mathematics from Jaypee Institute of Information Technology following her postgraduate studies in the M.Tech Computer Science. With extensive experience in teaching and research, Dr. Kaur has made significant contributions to the advancement of fuzzy automata theory and its applications. Her research interests include approximate string matching, fuzzy automata extensions, state minimization, multiset theory, and semantic computing, with a particular focus on practical applications such as resource allocation and air quality index prediction.
