Generative Fuzzy System for Sequence-to-Sequence Learning via Rule-based Inference

Hailong Yang, Zhaohong Deng, *Senior Member, IEEE*, Wei Zhang, Zhuangzhuang Zhao, Guanjin Wang, Kup-sze Choi, *Senior Member, IEEE*

***Abstract*—Generative Models (GMs), particularly Large Language Models (LLMs), have garnered significant attention in machine learning and artificial intelligence for their ability to generate new data by learning the statistical properties of training data and creating data that resemble the original data. This capability offers a wide range of applications across various domains. However, the complex structures and numerous model parameters of GMs obscure the input-output processes and complicate the understanding and control of the outputs. Moreover, the purely data-driven learning mechanism limits GMs' abilities to acquire broader knowledge. There remains substantial potential for enhancing the robustness and generalization capabilities of GMs. In this work, we leverage fuzzy system, a classical modeling method, to combine both data-driven and knowledge-driven mechanisms for generative tasks. We propose a novel Generative Fuzzy System framework, named GenFS, which integrates the deep learning capabilities of GMs with the term-based interpretability and dual-driven mechanisms of fuzzy systems. Specifically, we propose an end-to-end GenFS-based model for sequence generation, called FuzzyS2S. A series of test studies were conducted on 12 datasets, covering three distinct categories of generative tasks: machine translation, code generation, and summary generation. The results demonstrate that FuzzyS2S outperforms the Transformer in terms of accuracy and fluency. Furthermore, it exhibits better performance than state-of-the-art models T5 and CodeT5 for some application scenarios.**

***Index Terms*—Generative Model; Generative Fuzzy System; Sequence-to-Sequence; Transformer; Tokenizer**

I. INTRODUCTION

In recent years, generative models have garnered widespread attention for addressing complex generative tasks. Particularly, continuous development of Transformer and its derivative technologies has led to state-of-the-art large natural language processing models (LLMs), such as ChatGPT and Llama [1], [2], [3]. These models can explore hidden patterns and relationships within data and generate high-quality multimodal data such as text, audio, and image, making them powerful tools in Natural Language Processing (NLP) and artificial intelligence.

Transformer [4] and their derivatives form the basis of many generative AI techniques [5]. Representative models include GPT [6], [7], [8] for text generation, CLIP [9], [10] for image generation, and MuLan [11] for audio generation. Despite the complexity and variety of generative tasks, they can be transformed into sequence generative tasks through serialization techniques [12]. For example, text is sliced into token sequence, images are divided into patches, and audio data are discretized into time series of amplitudes. The models generate predicted sequences by learning the relationships between the input and target sequences [13]. These sequences are then deserialized to obtain the final results. Sequence generative tasks have several key characteristics, including variable-length unstructured data, ordered elements of sequence, and complex mapping relationship between input and target sequences.

Existing generative models have complex network structures, deep hierarchies, and a large number of parameters, which obscure the internal workings and decision-making processes and make output control difficult [14]. Also, these models often require vast datasets for training and operate as black boxes, and it is hard to incorporate logical knowledge, rules and constraints. This data-driven approach limits the models' ability to generalize and handle a wider range of applications. Generative models also demand significant computational time and resources for training. Moreover, it is challenging to reuse the parameters of the trained model, leading to considerable consumption of waste of computational resources, which hinders large-scale application and technological development in the long run.

Fuzzy system is a classical modeling method composed of fuzzy sets, fuzzy rules, and inference mechanisms. It handles fuzzy information effectively and provides good interpretability [15]. Also, fuzzy systems can express expert knowledge, address nonlinear problems, and offer better robustness and generalization capabilities. Notably, fuzzy systems can capture the fuzzy characteristics of human thinking from a macroscopic perspective, simulating human reasoning and decision-making to handle uncertainty problems that conventional mathematical methods struggle to solve. Fuzzy systems have been widely used for classification [16], [17], recognition [18], [19], detection [20], prediction and process modeling [21]. In fuzzy systems, fuzzy rules can represent priori expert knowledge, which can be easily

This work was supported in part by the National Key R&D Program of China under Grant 2022YFE0112400, and in part by the National Natural Science Foundation of China under Grant 62176105. (Corresponding author: Zhaohong Deng).

H. Yang, Z. Deng, W. Zhang and Z. Zhao are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China, and Engineering Research Center of Intelligent Technology for Healthcare, Ministry of Education, Wuxi 214122, China. (e-mail: yanghailong@stu.jiangnan.edu.cn; dengzhaohong@jiangnan.edu.cn; 7201607004@stu.jiangnan.edu.cn; zhaozhuangzhuang123@outlook.com).

G. Wang is with the School of Information Technology, Murdoch University, WA, Australia (e-mail: Guanjin.Wang@murdoch.edu.au).

K. S. Choi is with the TechCosmos Ltd., Hong Kong. (e-mail: kschoi@ieee.org).

transferred from one system to another [22], [23]. This enables efficient knowledge migration and reuse, greatly saving computational resources.

Fuzzy system uses a priori expert knowledge to divide the inputs into multiple sets with different feature terms for respective processing. Therefore, it is significant to develop generative models with fuzzy system. However, for complex generative tasks, such as machine translation, code generation, and summary generation in NLP, generative fuzzy systems still face the following challenges. Frist, the input of generative tasks consists of variable-length unstructured data [24], which cannot be directly processed by classical fuzzy systems. The complex token mapping relationships in generative tasks require models to be robust in handling high-dimensional data [25]. However, classical fuzzy systems are shallow models with a small number of trainable parameters, which are inadequate for running generative tasks effectively.

To address the first challenge, we develop generative fuzzy system by implementing a delegate election strategy for fuzzy sets and transforming the fuzzy membership calculation into similarity calculation between the inputs and the delegates. This approach enables fuzzification of both structured and unstructured data. Generative tasks are then performed by modeling the joint probability through learning the probability distribution of tokens across the sequences. The model needs to handle a large number of trainable parameters and has strong learnability. To address this challenge, we introduce deep generative models as the consequents of the fuzzy rules. This integration enhances the system's learning ability for sequence generation.

Compared with popular generative modeling approaches [26], [27], [28], the proposed generative fuzzy system enhances model performance and generalization by embedding prior knowledge through fuzzy rules. It retains the core advantages of classical fuzzy systems – term-based interpretability and dual-driven mechanisms powered by both expert knowledge and data-driven learning [29]. The contributions of this paper are summarized as follows. First, we propose a novel generative fuzzy system framework, called GenFS, which combines the high interpretability of fuzzy system with the powerful learning capabilities of generative models. Moreover, GenFS can efficiently transfer generative knowledge from one system to another through generative fuzzy rules, thereby saving computational resources and reducing training time costs.

Second, based on our proposed GenFS framework, we introduce a specific generative fuzzy system for natural language sequence generation, called FuzzyS2S. During the preprocessing stage, FuzzyS2S employs a novel multi-scale fuzzy tokenizer to optimize the token frequency distribution of the sequences and to extract sequence information at multiple scales. In addition, the model introduces an innovative fuzzy membership calculation method to effectively address the problem of variable-length sequence fuzzification.

The remainder of the paper is structured as follows: Part II introduces the generative model along with the related concepts and principles of classical fuzzy systems. Part III presents the generative fuzzy system framework GenFS. Part IV presents the generative fuzzy system FuzzyS2S for sequence-to-sequence generative tasks. Part V evaluates the performance of the FuzzyS2S model on machine translation, summary generation, and code generation. Part VI gives the conclusion and an outlook for future work.

II. Related Works

A. Generative Models

The advent of deep learning has led to significant advancements in generative modeling technology. In the field of NLP, sequence generation encompasses a range of sequence-to-sequence (seq2seq) [30] tasks such as machine translation, code generation, and summary generation. Early research primarily focuses on Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks [31]. However, the memory performance of these early seq2seq models degrades rapidly as sequence length increases, and they struggle to effectively differentiate tokens with varying importance [30].

To address these issues, Bahdanau et al. proposed the Attention Mechanism [32], which assigns different degrees of attention to tokens based on their importance by attentional scoring to alleviate the problem of long dependencies. Vaswani et al. proposed the Transformer model which utilizes a multi-head attention mechanism and effectively parallelizes training. BERT [33] is a model based on the Transformer architecture that employs bidirectional encoding representation, and is used as the encoder in the autoencoder framework [34]. To support a variety of downstream tasks, Raffel et al. proposed the T5 [35], a Transformer-based pre-trained language model capable of performing tasks such as text classification, text generation, text summarization, and machine translation. ChatGPT is based on the GPT (Generative Pre-trained Transformer) family, including GPT-3.5 and GPT-4 [36], which are capable of a wide range of downstream tasks and enable interactive conversations.

Although current generative models have achieved remarkable performance, it is important to note that their performance improves at the expense of exponentially grown parameter scale. These parameters are not readily reusable for other models [37], [38], [39], and it becomes increasingly challenging to interpret the results [40], [41]. Additionally, the learning processes of mainstream generative models are predominantly data-driven, lacking knowledge-driven mechanisms. Therefore, it is highly beneficial to develop a framework that can enhance the interpretability of generative models and embrace both data and knowledge driven mechanisms. For example, Zhong et al. proposed the E2S2 framework [42] to enhance the overall performance of encoder-decoder models by incorporating more effective self-supervised information into the encoders.

B. Fuzzy System

In 1965, American automatic control expert L.A. Zadeh proposed the concept of fuzzy sets, which led to the rapid development of fuzzy theory. Fuzzy system defines input, output and state variables on fuzzy sets. It is a representative uncertainty reasoning system and excels in solving nonlinear modeling problems. Fuzzy systems are now widely used in automatic control, pattern recognition, decision analysis, time series signal processing, among other tasks. Currently, there

are two main branches of fuzzy systems: Mamdani fuzzy systems and TSK (Takagi-Sugeno-Kang) fuzzy systems. TSK fuzzy system, in particular, has strong data-driven learning ability and good interpretability, which has garnered extensive attention in recent years.

Fuzzy reasoning in fuzzy system is approximate and non-deterministic, with both premises and conclusions being inherently fuzzy. By fuzzy reasoning, conclusions are derived from premises using hypothetical fuzzy propositions, known as fuzzy rules. A rule base is formed with multiple fuzzy rules. In the traditional and most widely used first-order TSK fuzzy system, the kth rule of its fuzzy rule base, i.e., the hypothetical fuzzy proposition IF-THEN rule, is represented as shown in (1). In this representation, the IF part constitutes the fuzzy rule antecedent, and the THEN part constitutes the fuzzy rule consequent.

$$\begin{gathered} IF\ x_1\ is\ A_1^k\ \wedge x_2\ is\ A_2^k\ \cdots \wedge\ x_D\ is A_D^k, \\ THEN\ f^k(x) = p_0^k + p_1^k x_1 + \cdots + p_D^k x_D \\ k = 1,2,3,\cdots,K \end{gathered} \tag{1}$$

In the above equation, D is the dimension of the sample feature space, K is the number of rules, $x_j (j = 1, 2, 3, \dots, D)$ is the jth feature of the input vector $\boldsymbol{x}$, and A_j^k is the antecedent fuzzy set of the kth fuzzy rule on the jth feature of the input vector $\boldsymbol{x}$. The symbol $\wedge$ denotes the fuzzy conjunction operator. The function $f^k(.)$ is the consequent processing function of the kth rule, which, in this context, is a classical linear function used to derive the output of the kth rule. The parameter $p_j^k (j = 1, 2, 3, \dots, D)$ is the jth parameter of the linear function adopted by the kth rule consequent. Construction of the antecedents and consequents of the TSK fuzzy system is detailed in Part 1 of the *Supplementary Materials*.

III. Generative Fuzzy System

A. Concept and Structure

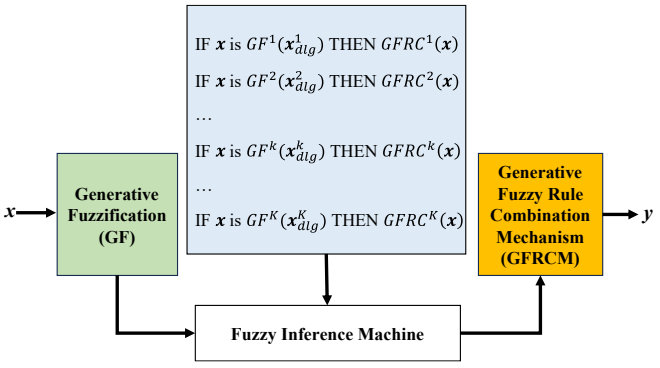


Fig. 1. Structure of Generative Fuzzy System Framework (GenFS)

Definition 1: Generative Fuzzy System Framework (GenFS). GenFS consists of a generative fuzzification module, a generative fuzzy rule base, a fuzzy inference machine, and a generative fuzzy rule combination mechanism, as illustrated in Fig. 1. GenFS operates by processing input data through the generative fuzzification module. Reasoning is then performed with generative fuzzy rules to draw fuzzy inference conclusions under given preconditions. These inference conclusions are integrated through an efficient generative fuzzy rule combination mechanism to yield the system's output. GenFS is a novel framework that combines generative models and classical fuzzy systems, enabling fuzzy systems to handle complex generative tasks.

Definition 2: Generative Fuzzification (GF). It is the process of delegate election of fuzzy sets to transform data into a more representative form and calculate the similarity between the inputs and the delegates.

Definition 3: Generative Fuzzy Rule Consequent (GFRC). GFRC is an intelligent fuzzy rule consequent that employs generative models as the consequent processing unit to enhance learning ability. GFRC can be represented as follows:

$$\boldsymbol{g}^k = GFRC^k(\boldsymbol{x}) \tag{2}$$

where $k = 1,2,3, \dots, K$, K is the number of rules in the fuzzy system, $GFRC^k$ is the generative rule consequent processing unit of the kth rule, and g^k is the generative data of the kth rule.

Definition 4: Generative Fuzzy Rules. Generative Fuzzy Rules are composed of antecedents with Generative Fuzzification and GFRCs. The kth generative fuzzy rule of the rule base can be expressed as follows:

$$IF\ \boldsymbol{x}\ is\ GF(\boldsymbol{x}_{dlg}^k)\ , THEN\ \boldsymbol{g}^k = GFRC^k(\boldsymbol{x}) \tag{3}$$

where $\boldsymbol{x}$ is the input data and $GF(\boldsymbol{x}_{dlg}^k)$ is the generative fuzzification function corresponding to the kth rule antecedent, represented by $\boldsymbol{x}_{dlg}^k$.

Definition 5: Generative Fuzzy Rule Combination Mechanism (GFRCM). GFRCM is an intelligent decision-making mechanism for generative tasks. Its purpose is to select applicable rule combination strategies to produce crisp and unambiguous results for different generative scenarios. When processing structured data with aligned features, GFRCM employs the weighted sum method to combine the outputs of all the rules. For unstructured output data, GFRCM uses the maximum defuzzification method, selecting the output of the rule with the highest fire strength as the final result. Eq. (4) provides the generalized form of generative fuzzy rule combination mechanism:

$$\boldsymbol{y} = GFRCM(\boldsymbol{G}) = \begin{cases} \sum_{k=1}^{K} \tilde{\mu}^k \boldsymbol{g}^k\ , \boldsymbol{G}\ is\ aligned \\ \boldsymbol{g}^{Argmax(\tilde{\boldsymbol{\mu}})}, otherwise \end{cases} \tag{4}$$

where $\boldsymbol{G} = \{g^1, g^2, g^3, \dots, g^K\}$, $\boldsymbol{g}^k$ is the output of kth rule, $\tilde{\mu}^k$ is the fire strength of the kth rule, and $\tilde{\boldsymbol{\mu}} = [\tilde{\mu}^1, \tilde{\mu}^2, \tilde{\mu}^3, \dots, \tilde{\mu}^K]$. When $\boldsymbol{G}$ is structured and the features are aligned, the fire strengths are used as weights to weight and sum all the rule outputs. Otherwise, the rule output with the maximum fire strength is selected as the final output, where $Argmax(\tilde{\boldsymbol{\mu}})$ function is used to obtain the index value of the maximum fire strength.

B. Construction of Generative Fuzzy Rule Antecedents

The construction of generative fuzzy rule antecedents is a crucial task in modeling generative fuzzy systems. Classical fuzzy systems use the membership of the crisp data structured

features corresponding to all the fuzzy sets as the fuzzification results. However, the inputs of generative tasks often vary in length after serialization, classical fuzzification methods are inadequate for directly handling unstructured data. Therefore, a novel generative fuzzification method for complex generative tasks is developed.

Membership calculation in generative fuzzification method needs to solve two key problems – election of delegates for fuzzy sets and calculation of similarity between the inputs and the delegates.

Delegate Election. A delegate of a set is a specific element of the set [43], [44]. The election of a delegate element simplifies the understanding and manipulation of the structure and properties of the set. In generative tasks, the delegate of a fuzzy set is a specific sample within the set that has a high degree of similarity in structure and properties with other samples. The way of delegate election can be based on expert experience or methods such as clustering [45]. Each fuzzy set has its own individual delegate. The election of delegates can be expressed as

$$\boldsymbol{X}_{dlg} = DSM(\boldsymbol{X}, K) \tag{5}$$

where $\boldsymbol{X}_{dlg} = \left[x_{dlg}^1, x_{dlg}^2, \dots, x_{dlg}^K\right]$ is a delegate set of K rules in the system rule base, $\boldsymbol{X}$ is the set of corresponding input data, DSM denotes the method of the delegate election of the fuzzy set $\boldsymbol{X}$.

Similarity Calculation. Similarity is expressed as a numerical value where a larger value indicates greater similarity between two objects. In generative tasks, the inputs and delegates are usually unstructured data with unaligned features. Therefore, it is necessary to design flexible calculation methods that can be applied to different scenarios. Similarity can be calculated by data matching, feature aligning, and other specific methods. It can be expressed as the generalized form,

$$\mu^k = Sim\left(\boldsymbol{x} | \boldsymbol{x}_{dlg}^k\right) \tag{6}$$

where $\boldsymbol{x}$ is the input data, $\boldsymbol{x}_{dlg}^k$ is the delegate of the kth fuzzy set, $k = 1,2,\dots,K$, K is the number of rules, $Sim(.)$ is a generalized method for calculating the similarity, μ^k is the similarity between $\boldsymbol{x}$ and the delegate of the corresponding fuzzy set of the kth rule, namely, the fire strength of $\boldsymbol{x}$ about the kth rule.

To stabilize numerical calculations and avoid feature bias, it is usually necessary to normalize similarity, i.e.,

$$\tilde{\boldsymbol{\mu}} = Norm(\boldsymbol{\mu}) = \frac{\boldsymbol{\mu}}{\sum_{k=1}^{K} \mu^k} \tag{7}$$

where $\tilde{\boldsymbol{\mu}} = [\tilde{\mu}^1, \tilde{\mu}^2, \tilde{\mu}^3, \dots, \tilde{\mu}^K]$, $\boldsymbol{\mu} = [\mu^1, \mu^2, \mu^3, \dots, \mu^K]$.

While random seeds may introduce slight variations in the delegates generated by clustering, these delegates remain highly similar and effectively representative of their domain-specific sets. As a result, the impact on overall model performance is negligible.

C. Learning and Optimization of GFRCs

The learning of GFRCs can be regarded as a typical machine learning task. In machine learning, the closer the predicted data distribution is to the real data distribution, the better the model's performance. In generative task, calculating the difference between the target sequence and the generated sequence is key to the model's learning process. From a probability distribution perspective, the target sequence is considered as the conditional probability distribution of the input sequence, and the difference between the generated sequence and the target sequence is derived from these two probability distributions. The cross-entropy loss function is widely used to measure this probability distribution difference. From a sequence similarity perspective, the difference between sequences can be measured by their similarity, so that the similarity loss function is also applicable for sequence difference calculation. Depending on the scenarios, GenFS adopts different loss functions to calculate sequence differences. To prevent overfitting, an L2 regularization term is added to the loss function. The final loss function is generalized as follows:

$$\mathcal{L}_{GFRC}(\Theta) = diff(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \gamma \|\Theta\|_2 \tag{8}$$

where $\boldsymbol{y}$ is the ground truth for the target sequences in the generative task, $\hat{\boldsymbol{y}}$ is the generative sequence, $diff(.)$ is a function to compute the difference between the target and generative sequences, $\|\Theta\|_2$ is the L2 regular term of the GFRC consequent parameters, and $\gamma(\gamma > 0)$ is a balancing parameter.

For generative tasks, sequence data is high-dimensional and sparse, with an uneven token frequency distribution where high-frequency and low-frequency tokens appear randomly. During training, reducing the learning rate for high-frequency tokens and increasing it for low-frequency tokens helps the model to learn information effectively from both types of tokens. Therefore, GFRC needs to employ optimizers capable of adaptively adjusting the learning rate, such as AdaGrad, RMSprop, Adadelta and Adam [46], [47], to better approximate the optimal values of the model parameters and minimize the training loss. The generalized representation of the adaptive optimizer is

$$\Theta' = AdaOptim(\Theta, \ell_s) \tag{9}$$

where $AdaOptim(.)$ is the adaptive optimizer function, ℓ_s is the initialised value of the learning rate, and Θ' is the model parameter that has been computationally updated by the optimizer.

All subdomain representatives are obtained using automated clustering methods. These representatives are associated with a set of domain-specific terms and are used to construct term-based expressions in the rule antecedents, while the rule consequents are modeled using generative models. This structure allows GenFS to embed domain-specific prior knowledge and expert experience through antecedents, while the consequents capture hidden patterns from data via learning by combining antecedents and consequents. Hence, GenFS achieves a dual-driven mechanism leveraging both knowledge and data. To incorporate additional expert rules or domain constraints into GenFS, the corresponding domain representatives must be provided to construct generative fuzzy rules. These rules are seamlessly integrated into the GenFS framework to significantly enhance both the term-based interpretability and controllability of GenFS-based models.

D. Generalized Learning Algorithm for GenFS

In knowledge-based systems, a series of IF-THEN conditional statements can be used as rules to express knowledge. The parameters of the antecedents can be derived from the prior knowledge of human experts or clustering techniques. The consequents acquire knowledge by learning the mapping relationship of the sequences. The construction of generative fuzzy rules involves the construction of the antecedents and the learning the consequents' parameters.

The model parameters in the generative fuzzy system, denoted as Θ, include the parameters of both the antecedents and consequents. The consequents can be initialized randomly or with pre-trained model parameters. Multiple consequents can be trained jointly. The generalized learning algorithm for GenFS is described in Algorithm I.

Algorithm I: GenFS Learning

Input: input sequence $\boldsymbol{x}$, input sequence set $\boldsymbol{X}$, target sequence $\boldsymbol{y}$, target sequence set $\boldsymbol{Y}$, rule total K, maximal number of epochs E;
Output: model parameters Θ

Procedure:
// *antecedent generation*
1: cluster fuzzy set delegates $\boldsymbol{X}_{dlg} = [\boldsymbol{x}_{dlg}^1, \boldsymbol{x}_{dlg}^2, \dots, \boldsymbol{x}_{dlg}^K]$ from $\boldsymbol{X}$ based on (5)
// *consequent learning*
2: **for** $e = 1, .., E$ **do**
3: **for** $\boldsymbol{x}$ in $\boldsymbol{X}$ **do**
4: **for** $k = 1..K$ **do**
5: calculate rule strength μ^k based on (6)
6: calculate rule consequent result g^k
7: **end for**
8: strength normalization $\tilde{\boldsymbol{\mu}}$
9: get $\hat{\boldsymbol{y}}$ through rule combination based on (4)
10: $\mathcal{L}_{GFRC}(\Theta) = diff(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \gamma\|\Theta\|_2$
11: $\Theta \coloneqq \underset{\Theta}{\operatorname{argmin}} \mathcal{L}_{GFRC}$, optime model parameters Θ
12: **end for**
13: **end for**
14: return Θ

IV. Sequence-to-Sequence Learning Based On GenFS Framework

In this section, we propose a specific generative fuzzy system based on the GenFS framework, tailored for sequence-to-sequence generation tasks. First, we define the specific problems involved. Then, we propose a novel multi-scale tokenizer, namely fuzzy tokenizer, for sequence splitting preprocessing. Finally, we propose an end-to-end generative model based on GenFS, named FuzzyS2S.

A. Definition of Sequence Generation Problems

In NLP tasks such as machine translation, code generation, and summary generation, the text sequence is sliced into tokens by tokenizers, and these tokens follow the long-tailed distribution of Zipf's Law [48]. Zipf's Law is an empirical law that describes the relationship between token frequency and token ranking in natural language. The basic formulation of Zipf's Law is that in a large corpus, the token frequency is inversely proportional to its position in the token ranking list. Specifically, the law is expressed as:

$$f(n) \approx C/n \tag{10}$$

where $f(n)$ is the frequency of the token in the ranking list, n is the rank of the token, and C is a constant.

In natural language, token occurrences typically follow Zipf's Law, where a small number of tokens appear with high frequency while the majority occur infrequently. In real-world sequence generation, this distribution gives rise to several persistent challenges. Notably, significant variations in sequence lengths and token frequency distributions complicate model fitting [49], while the inherent ambiguity and uncertainty within sequences undermine model robustness and generalization capabilities [50]. These characteristics lead to two key problems below that warrant focused investigation.

Problem I: Low-Frequency Underfitting

In sequence modeling tasks, a substantial proportion of tokens appearing at low frequency would result in sparse occurrence patterns that hinder the model's ability to learn and generalize their behaviors effectively. This problem is known as in low-frequency underfitting, where the model struggles to capture the semantics and structural roles of rare tokens, thereby degrading overall performance. Addressing this issue involves two key strategies:

(1) Reduce the number of low-frequency tokens through techniques such as token merging, clustering, or vocabulary optimization.

(2) Optimize the overall token distribution to enhance the learnability of token occurrence patterns, particularly for rare events, thus improving the model's performance across diverse sequences.

Problem II: Significant Variations in Token Distributions and Sequence Lengths

Sequences in real-world generative tasks often exhibit considerable variability in both token distributions and sequence lengths. The heterogeneity poses additional challenges for effective model fitting and generalization:

(1) Variations in token frequency distributions impair model's ability to generalize different types of sequences, particularly when their underlying statistical properties vary significantly.

(2) Variations in sequence lengths can lead to inefficient training and complicate the allocation of attention or memory resources, especially for models constrained by fixed or limited context windows.

Therefore, it is vital to build robust models capable of handling diverse and heterogeneous real-world sequence data effectively.

B. Multiscale Tokenizer based on the priori expert knowledge: Fuzzy Tokenizer

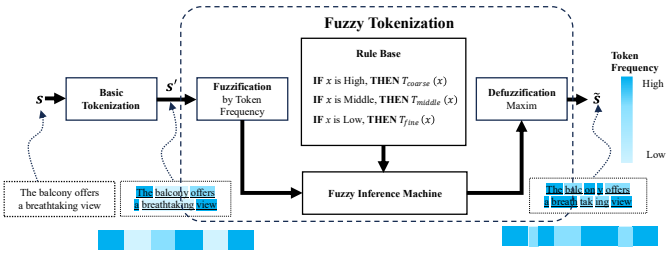


Fig. 2. Structure of Fuzzy Tokenizer

To address the issue of low-frequency tokens (Problem I), we propose a specific multi-scale tokenizer, called fuzzy

tokenizer, which is a fuzzy system based on multi-scale sub-word tokenizers [51] that enables adaptive slicing of words at different scales to optime the distribution of token frequency. The architecture of the fuzzy tokenizer is illustrated in Fig. 2.

Given the original text sequence $\boldsymbol{s}$, a preliminary token sequence $\boldsymbol{s}'$ can be obtained as follows after the basic tokenization,

$$\boldsymbol{s}' = T_{basic}(\boldsymbol{s}) = [x_1, x_2, x_3, \dots, x_N] \tag{11}$$

where T_{basic} is the basic tokenizer, which splits text sequences with spaces or punctuation marks as separators. n is the total number of tokens of the sequence $\boldsymbol{s}'$, x_n, $n = 1,2,3, \dots, N$ is the nth token of the sequence $\boldsymbol{s}'$.

In fuzzy systems, commonly used membership functions include triangular, trapezoidal, and Gaussian functions [52]. Among them, Gaussian function offers superior smoothness and differentiability. It captures the representativeness of high-frequency terms more accurately while effectively suppressing the influence of long-tail tokens inherent in Zipf distributions. Therefore, Gaussian function is adopted in this study for the antecedents of the rules to calculate the similarity between the input tokens and the delegates of the fuzzy sets. The similarity is used to represent the fire strength of the fuzzy rules. The fire strength $\tilde{\mu}_n$ of token x_n with respect to all rules can be expressed as follows:

$$\mu_n^k = Sim_g\left(x_n, x_{dlg}^k\right) \tag{12}$$

$$\tilde{\boldsymbol{\mu}}_n = \frac{\boldsymbol{\mu}_n}{\sum_{k=1}^{K} \mu_n^k} \tag{13}$$

where μ_n^k is the fire strength of token x_n with respect to the kth rule, i.e., $\boldsymbol{\mu}_n = [\mu_n^1, \mu_n^2, \mu_n^3, \dots, \mu_n^K]$, and the normalised fire strength $\tilde{\boldsymbol{\mu}}_n = [\tilde{\mu}_n^1, \tilde{\mu}_n^2, \tilde{\mu}_n^3, \dots, \tilde{\mu}_n^K]$, Sim_g is the Gaussian-based similarity calculation function for tokens.

The consequents of the fuzzy tokenizer are sub-word tokenizers operating at different scales. The outputs are unstructured sub-word sequences. Fuzzy tokenizer selects the maximum defuzzification method to obtain the final result as follows,

$$\tilde{\boldsymbol{x}}_{\boldsymbol{n}} = T_{fuzz}^{Argmax(\tilde{\boldsymbol{\mu}}_{\boldsymbol{n}})}(x_n) \tag{14}$$

where $\tilde{\boldsymbol{x}}_{\boldsymbol{n}} = \left[x_{n,1}, x_{n,2}, x_{n,3}, \dots, x_{n,\tau}\right]$ is the sub-word sequence after slicing of token x_n and τ is the total number of sub-words. The fuzzy tokenizer splices the results of further slicing to obtain the final token sequence $\tilde{\boldsymbol{s}}$ as follows.

$$\tilde{\boldsymbol{s}} = concat(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \tilde{\boldsymbol{x}}_3, \dots, \tilde{\boldsymbol{x}}_N) \tag{15}$$

When there are multiple rules with similarly high activation, the tokenizer with the coarsest granularity is selected to minimize the generated sequence length. Most tokenizers include four special tokens: *bos* (beginning of sequence), *eos* (end of sequence), *pad* (padding), and *unk* (unknown). For native special symbols in the text, such as @, #, $, %, /, -, _, &, we deliberately bypass the multi-scale tokenization process and simply handle them with basic tokenization. This prevents noisy tokens and ambiguity, thereby avoiding degradation in system performance.

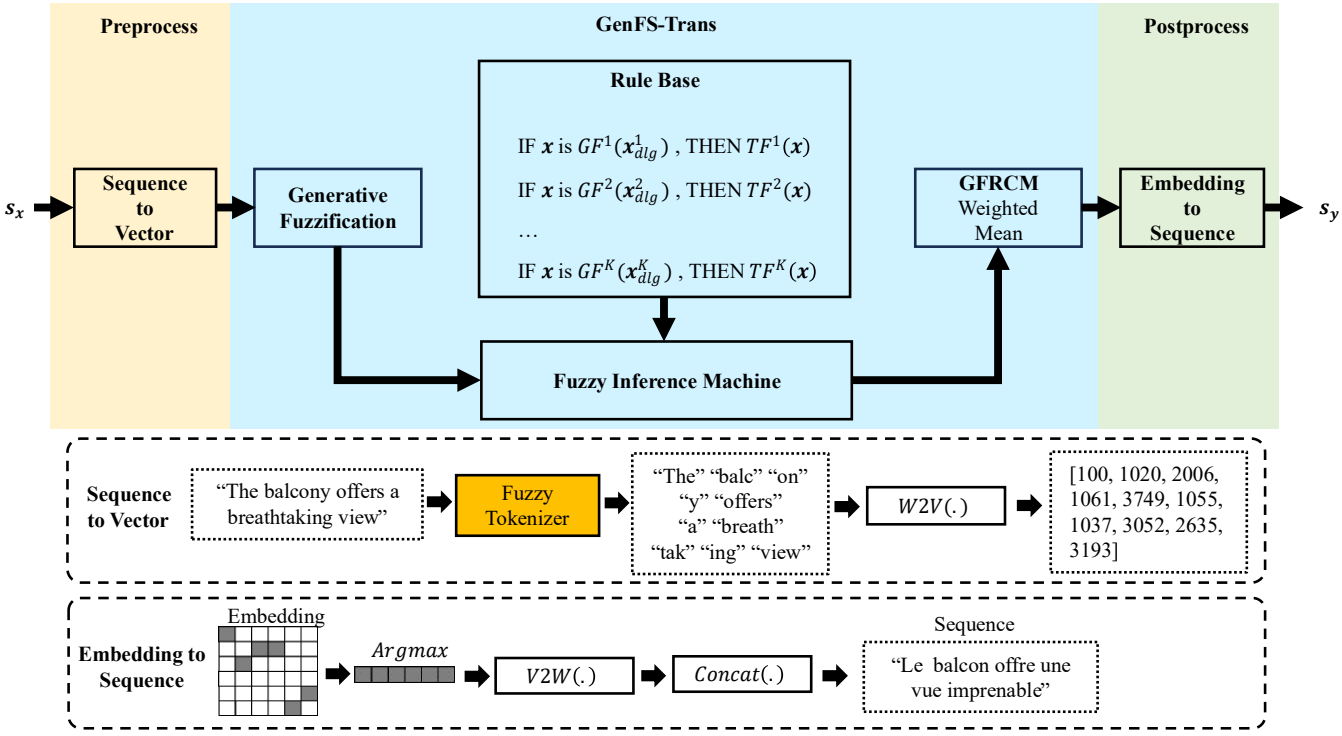


Fig. 3. Structure of FuzzyS2S. TF^k is the Transformer processing unit of the kth rule consequent, $\boldsymbol{s}_x$ is the input sequence, and $\boldsymbol{s}_y$ is the target sequence. The Preprocess module named *Sequence to Vector* is to implement the conversion from sequences to word vectors, and the Postprocess module named *Embedding to Sequence* is to convert the decoded word embeddings into the target sequences.

By combining fuzzy system and multi-scale learning, the fuzzy tokenizer is designed to obtain more optimal splitting results based on fuzzy inference, considering the inherent fuzziness and uncertainty of the tokens at different scales. By integrating sub-word tokenizers of different scales, the fuzzy tokenizer leverages the universal approximator properties of fuzzy system [53], [54], [55], [56] to approximate the token sequences with the best token frequency distribution and suitable sequence length.

C. Sequence-to-sequence GenFS: FuzzyS2S

This section presents the end-to-end sequence generative model FuzzyS2S which is based on GenFS. In FuzzyS2S, we implement a special GenFS using Transformer as the consequent processing units, denoted as GenFS-Trans. FuzzyS2S consists of three modules – Preprocess module, GenFS-Trans module and Postprocess module, which are described below

Preprocess Module. This module transforms the original input sequences into vectors, reducing the low-frequency tokens in the sequences using the fuzzy tokenizer. The source sequence $\boldsymbol{s_x}$ is passed through the fuzzy tokenizer to obtain the vector $\boldsymbol{v_x}$ as follows,

$$\boldsymbol{v_x} = W2V(T_{fuzzy}(\boldsymbol{s_x})) \tag{16}$$

where $W2V$ represents the method that maps the token sequences to vectors.

The tokens sliced by T_{fuzzy} are saved using the classical bag-of-words (BoW) model [57], [58], [59]. The input of BoW is the sequence of tokens, and the output is the word vector, which is an array of indexes of the storage locations, and the size N_{bag} of the BoW. The BoW contains two special tokens, the start symbol *bos* and the end symbol *eos*, which are added at the start and end positions of the sequence, respectively.

To construct the antecedents of generative rules, K delegates are selected from the vector set as follows,

$$\left[\boldsymbol{v}_{dlg}^{1}, \boldsymbol{v}_{dlg}^{2}, \dots, \boldsymbol{v}_{dlg}^{k}, \dots, \boldsymbol{v}_{dlg}^{K}\right] = DSM_{FCM}(\boldsymbol{V}_x, K) \tag{17}$$

where $\boldsymbol{V}_x$ is the set of input word vector, $DSM_{FCM}(.)$ is a delegate election method based on Fuzzy C-Means (FCM) [60], the primary features considered in FCM-based unsupervised clustering include sequence length and the frequency of high-occurrence tokens. The fuzzy set delegate is obtained by unsupervised fuzzy clustering.

GenFS-Trans Module. This module is the core of FuzzyS2S. It begins by performing fuzzification on the input vectors, then conducts the fuzzy inference using the fuzzy inference machine, and finally fuses the results of all the rules via the GFRCM component. The similarity between the inputs and the delegates are computed using methods such as Euclidean distance [61], Manhattan distance [62], cosine similarity [63] and Jaccard Similarity. Among these, cosine similarity is commonly adopted, particularly for large-scale text processing tasks, due to its efficiency and effectiveness in capturing textual similarity [64]. The formulations involved in the module are shown in (18) to (23). Refer to (18), we employ cosine similarity to measure the similarity between an input sequence and its corresponding set representative. The consequent processing units in our framework are based on Transformer. While the interpretability of GenFS primarily derives from the explainable, term-based representations in its antecedents, the consequents remain inherently opaque due to the black-box nature of Transformers. According to (19), the word vectors are transformed into word embeddings through positional encoding to preserve the positional information of the token in the sequence. (20) shows that the word embeddings output by Transformer are passed through a multilayer perceptron (MLP) to raise the feature dimensions to N_{bag}. Furthermore, in the GFRCM component of GenFS-Trans, the weighted sum method is used to combine the results of all the rules. The fire strengths of the rules serve as weights, and the word embeddings are weighed to obtain the target word embedding $\boldsymbol{Em}_{\hat{y}}$ as shown in (22). Finally, the resulting embedding is compressed into the range of (0,1) to obtain the $\boldsymbol{Em}_{\tilde{y}}$ matrix through $Softmax$ function in (23).

$$\mu_s^k = Sim_{cos}\left(\boldsymbol{v_x} | \boldsymbol{v}_{dlg}^k\right) \tag{18}$$

$$\boldsymbol{Em}_x = Pos\left(Emb\left(B(\boldsymbol{v_x})\right)\right) \tag{19}$$

$$\boldsymbol{Em}_{\hat{y}}^k = MLP\left(TF^k(\boldsymbol{Em}_x)\right) \tag{20}$$

$$\tilde{\boldsymbol{\mu}}_s = \frac{\boldsymbol{\mu}_s}{\sum_{k=1}^{K} \mu_s^k} \tag{21}$$

$$\boldsymbol{Em}_{\hat{y}} = \sum_{k=1}^{K} \tilde{\mu}_s^k \boldsymbol{Em}_{\hat{y}}^k \tag{22}$$

$$\boldsymbol{Em}_{\tilde{y}} = Softmax(\boldsymbol{Em}_{\hat{y}}) \tag{23}$$

where the cosine distance algorithm $Sim_{cos}(.)$ is used to calculate the similarity between the input data and the delegates; $\boldsymbol{\mu}_s = [\mu_s^1, \mu_s^2, \mu_s^3, \dots, \mu_s^K]$, normalized similarity $\tilde{\boldsymbol{\mu}}_s = [\tilde{\mu}_s^1, \tilde{\mu}_s^2, \tilde{\mu}_s^3, \dots, \tilde{\mu}_s^K]$; v_x is the word vector of the source sequence, $B(.)$ is to add *bos* at the beginning of the word vector, $Emb(.)$ is to compute the word embedding of the word vector, $Pos(.)$ is to encode the position of the word embedding; TF^k is the Transformer processing unit for the kth consequent, $\boldsymbol{Em}_x \in \mathbb{R}^{(N+1)\times Dim}$, Dim is the dimensionality of the word embedding, $\boldsymbol{Em}_{\hat{y}} \in R^{(M+1)\times N_{bag}}$, $\boldsymbol{Em}_{\tilde{y}}$ is the predictive probability matrix, $\boldsymbol{Em}_{\tilde{y}} \in \mathbb{R}^{(M+1)\times N_{bag}}$, the ground true of the target embedding $\boldsymbol{Em}_y = Pos\left(Emb\left(E(\boldsymbol{v_y})\right)\right)$, $\boldsymbol{v_y}$ is the vector of the target sequence, $E(.)$ is to add *eos* at the end of the word vector.

Postprocess Module. This module is responsible for converting the word embeddings output from the GenFS-Trans module into the target sequence. In this module, the $Argmax$ function is used to obtain the index of the maximum value of the probability for each dimension, and $M+1$ tokens are then predicted. The process is expressed as

$$\begin{aligned} \boldsymbol{s}_{\hat{y}} &= Concat\left(V2W\left(Argmax(\boldsymbol{Em}_{\tilde{y}})\right)\right) \\ &= Concat([\hat{y}_1, \hat{y}_2, \hat{y}_3, \dots, \hat{y}_M, eos]) \end{aligned} \tag{24}$$

where $V2W(.)$ is a vector-to-token conversion function and $Concat(.)$ is a function that splices an array of tokens to a sequence, FuzzyS2S stops the token prediction when the stop marker *eos* is reached.

To clearly illustrate the computation and optimization of FuzzyS2S, the modeling process is presented in Algorithm II.

Algorithm II: FuzzyS2S Modeling Process

Input: source sequence $\boldsymbol{s_x}$, target sequence $\boldsymbol{s_y}$, number of rules K,maximal number of epochs E;
Output: model parameters Θ

Procedure:
// *preprocess*
Vectorizing $\boldsymbol{v}_x = W2V(T_{fuzzy}(\boldsymbol{s}_x)), \boldsymbol{v}_x \in \boldsymbol{V}_x$
Vectorizing $\boldsymbol{v}_y = W2V(T_{fuzzy}(\boldsymbol{s}_y)), \boldsymbol{v}_y \in \boldsymbol{V}_y$
Select K delegates $[\boldsymbol{v}_{dlg}^1 \ldots, \boldsymbol{v}_{dlg}^K]$ from $\boldsymbol{V}_x$ as (17)
// *GenFS-Trans*
for $e = 1..E$ **do** // *training epoch*
  for $\boldsymbol{v}_x, \boldsymbol{v}_y$ in $\boldsymbol{V}_x, \boldsymbol{V}_y$
    for $k = 1..K$ **do** // *rule calculation*
      calculate rule strength μ_s^k based on (18)
      calculate rule consequent, $\boldsymbol{Em}_{\hat{y}}^k$ based on (19-20)
    end for
    $\tilde{\boldsymbol{\mu}}_s = \frac{\boldsymbol{\mu}_s}{\sum_{k=1}^K \mu_s^k}$ //*rule strength normalization*
    $\boldsymbol{Em}_{\hat{y}} = \sum_{k=1}^K \tilde{\mu}_s^k \boldsymbol{Em}_{\hat{y}}^k$ //*rule combination*
    $\boldsymbol{Em}_{\tilde{y}} = Softmax(\boldsymbol{Em}_{\hat{y}})$
    $\boldsymbol{Em}_y = Pos\left(Emb\left(E(\boldsymbol{v}_y)\right)\right)$
    $\mathcal{L}_{GFRC}(\Theta) = diff(\boldsymbol{Em}_y, \boldsymbol{Em}_{\tilde{y}}) + \gamma\|\Theta\|_2$ based on (8)
    optimize model parameters $\Theta \coloneqq \underset{\Theta}{\operatorname{argmin}} \mathcal{L}_{GFRC}$
  end for
end for
// *postprocess*
the embedding $\boldsymbol{Em}_{\tilde{y}}$ to the sequence $\boldsymbol{s}_{\hat{y}}$ based on (24)
return Θ

Although the current GenFS-based FuzzyS2S model is designed specifically for text generation, the underlying GenFS framework is highly extensible. It has the potential for adaptation to cross-modal tasks like text-to-image and text-to-audio generation. To support multimodal outputs, GenFS must be extended to incorporate multimodal fuzzy rules along with mechanisms for inter-modal alignment. However, this is non-trivial due to challenges in modeling cross-heterogeneous output modalities and developing robust alignment strategies between them. Future work will focus on these challenges to extend GenFS for a broader range of multimodal generative tasks.

V. Tests and Analysis

A. Datasets and Test Setting

The experiments followed the Train-Validation-Test (TVT) approach as outlined in [65]. The samples are divided into three parts: training sets, validation sets, and test sets at the ratio of 8:1:1 ratio, with 80% of the data for model training, 10% for validation during each training epoch, and the remaining 10% for testing the final model. We employed 10-fold cross-validation [66], repeating the entire experiment 10 times and reporting the averaged results.

Three types of data were used, including machine translation, summary generation, and code generation, totaling 12 datasets. The datasets for machine translation are WMT14 [65], Tatoeba [67] , EUconst [68] and Ubuntu [68]. The datasets for summary generation are CNN/DM (CNN Daily Mail) [69], [70], SAMSum [71], XLSum[72] and BillSum [73]. The datasets for code generation are HS (HearthStone) [74], MTG (Magic the Game) [74], GEO (Geoquery) and Spider [75]. Details of these datasets are provided in Part 2 of the *Supplementary Materials*.

The experiments were conducted on a computer equipped with an Intel i7 CPU (12 cores, 2.53 GHz), an NVIDIA RTX 4090 GPU (24 GB, 2.28 GHz), and 96 GB of RAM.

GenFS is a modeling framework based on generative fuzzy rules, in which the number of rules is the most critical hyperparameter. In the GenFS-based FuzzyS2S, we determine the number of rules using the "domain decomposition" as follows. First, we analyze the task to identify the major domains involved and based on the statistical descriptions of the dataset and expert knowledge of human experts or LLM-based agents, we further decompose each domain into several subdomains. If a subdomain remains overly complex, it can be recursively subdivided. The final number of subdomains determines the number of fuzzy rules required. By applying domain decomposition to the 12 datasets, the sequences can be categorized into three subdomains based on their lengths, i.e., long, medium, and short. Accordingly, the number of rules is set to 3 by default.

TABLE I
PARAMETERS, STRUCTURE AND COMPLEXITY OF BASELINES

Model	Param	Structure	FLOPs	GPU
(Code)T5-Small [35]	60M	6E+6D	1.81×10^{20}	240MB
(Code)T5-Base [35]	222M	12E+12D	6.62×10^{20}	888MB
(Code)T5-Large [35]	737M	24 E+24D	2.35×10^{21}	2.9GB
E2S2-T5[42]	745M	24E+24D	2.71×10^{21}	3.1GB
Transformer [4]	42M	3E+3D	1.26×10^{20}	168MB
AvgTrans	132M	3(3E+3D)	3.61×10^{20}	528MB
FuzzyS2S	137M	3(3E+3D)	3.63×10^{20}	548MB

*"E" denotes a Transformer encoder layer, while "D" denotes a decoder layer. E2S2-T5 is the model obtained by enhancing the encoders of T5-Large using the E2S2 framework.

To verify the advancement of FuzzyS2S more effectively, we designed a hybrid model with structure and computational complexity similar to FuzzyS2S, namely AvgTrans. It fused multiple Transformers through the averaged sum method. The comparative methods in the tests are Transformer, T5 family (T5, CodeT5 [76], E2S2-T5) AvgTrans (averaged sum) and FuzzyS2S (weighted sum). The model parameter sizes (Param), model structure (Structure), training floating-point operations (FLOPs), and GPU memory usage during training (GPU) for all methods are shown in Table I. It can be seen from Table I that the model structures of FuzzyS2S and AvgTrans are similar, both consisting of 3 Transformers. To ensure fairness in the comparisons, the T5 family models are retrained on the relevant datasets to eliminate potential biases arising from pretraining corpora or task-specific differences. Further details of these models can be found in Part 3(A) of the *Supplementary Materials*. The parameter settings for FuzzyS2S and the indicators for model evaluation, including ACC, BLEU [77], METEOR [78], ROUGE-1(R1), ROUGE-2(R2), and ROUGE-L(RL), are detailed in Part 3(B) and 3(C) of the *Supplementary Materials*. The codes and model parameters of our FuzzyS2S are available at https://github.com/chinesebear/fuzzys2s.

B. Analysis of Sequence Generation Tests

1) Analysis of Machine Translation Tests

From the results in Table II, it can be seen that FuzzyS2S outperforms E2S2-T5 in terms of accuracy, 50.88 higher on the Tatoeba dataset and 52.10 higher on the EUconst dataset. Furthermore, the BLEU score is 11.03 higher and the METEOR score 1.91 higher than that of E2S2-T5 on the EUconst dataset. Although the performance of FuzzyS2S in terms of BLEU and METEOR scores are not always better than the models of the T5 family, FuzzyS2S still shows an overall advantage. The results of comparison can be analyzed as follows:

(1) FuzzyS2S enhances accuracy by leveraging the interconnectivity between tokens of different scales. When the occurrence frequency of coarse scale tokens is small, it becomes challenging for the model to capture their semantic information. By splitting the text sequence into fine-scale tokens, the occurrence frequency of these tokens can be increased.

(2) The generative consequents of FuzzyS2S are based on Transformer, which employs absolute positional coding [79]. In contrast, models of the T5 family employ relative positional coding [80]. However, the relative position coding considers the relative distance between the current token position and the position of the token under attention when calculating the attention score [80]. This approach focuses more on the fluency of the sequences, which can decrease the accuracy of the token prediction.

Above all, FuzzyS2S exhibits better performance than AvgTrans. This improvement can be attributed to the fusion of the GFRCs (Transformer) using the weighted sum method, with fire strengths as weights. This method is similar to the naive soft attention mechanism [81], [82], [83]. As fire strength can be changed adaptively based on the inputs, the weighted fusion method functions as a dynamically adaptive soft-attention mechanism. This attention mechanism enables FuzzyS2S to outperform Transformer in general.

TABLE II
RESULTS OF MACHINE TRANSLATION TESTS

Model	WMT14			Tatoeba			EUconst			Ubuntu		
	ACC	BLEU	METEOR	ACC	BLEU	METEOR	ACC	BLEU	METEOR	ACC	BLEU	METEOR
T5-Small	6.70±0.12	27.31±0.08	53.71±0.03	7.23±0.04	36.20±0.06	61.37±0.04	2.97±0.11	36.75±0.03	75.88±0.05	4.24±0.01	2.76±0.10	15.61±0.01
T5-Base	7.33±0.05	29.96±0.12	55.74±0.06	7.89±0.11	39.96±0.08	64.07±0.11	2.81±0.10	38.26±0.08	77.14±0.06	4.48±0.13	3.33±0.07	16.52±0.03
T5-Large	7.40±0.01	30.92±0.12	56.81±0.08	8.01±0.04	41.45±0.11	65.19±0.02	2.87±0.09	38.59±0.13	77.42±0.05	5.65±0.05	3.90±0.05	17.43±0.10
E2S2-T5	8.21±0.03	**31.54±0.04**	**57.80±0.05**	9.04±0.01	**42.13±0.07**	65.20±0.02	3.51±0.06	38.61±0.09	77.21±0.03	6.32±0.07	4.23±0.08	19.58±0.07
Transformer	7.71±0.11	6.15±0.05	31.9±0.11	59.91±0.07	37.80±0.09	59.01±0.05	55.45±0.10	37.72±0.04	69.88±0.08	14.56±0.11	7.97±0.10	41.24±0.04
AvgTrans	8.10±0.17	11.42±0.07	37.65±0.03	57.63±0.10	36.58±0.07	62.56±0.04	52.68±0.13	40.69±0.08	73.68±0.14	14.36±0.03	7.90±0.16	41.69±0.09
FuzzyS2S	**8.49±0.13**	11.20±0.10	39.36±0.09	**59.92±0.11**	37.83±0.13	**66.08±0.10**	**55.61±0.11**	**49.64±0.13**	**79.12±0.10**	**14.67±0.07**	**7.99±0.10**	**41.78±0.02**

TABLE III
RESULTS OF SUMMARY GENERATION TESTS

Model	CNN/DM			SAMSum			XLSum			BillSum		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
T5-Small	41.12±0.09	19.56±0.11	38.35±0.04	23.48±0.13	6.10±0.04	18.69±0.08	17.84±0.10	4.96±0.03	12.97±0.05	24.89±0.02	10.23±0.12	17.76±0.06
T5-Base	42.05±0.00	20.34±0.09	39.40±0.10	24.14±0.09	6.85±0.02	18.79±0.12	18.16±0.06	5.08±0.12	12.65±0.07	25.45±0.05	11.76±0.01	18.77±0.12
T5-Large	42.50±0.13	20.68±0.05	**39.94±0.07**	27.36±0.06	9.09±0.09	22.09±0.06	23.14±0.10	7.14±0.02	16.04±0.04	28.64±0.09	13.67±0.13	21.75±0.08
E2S2-T5	**42.75±0.01**	**21.34±0.01**	39.90±0.04	28.31±0.09	8.11±0.03	22.21±0.06	23.44±0.07	7.01±0.03	15.60±0.05	30.21±0.07	14.04±0.08	23.59±0.03
Transformer	24.51±0.02	4.51±0.06	18.08±0.04	30.16±0.03	6.03±0.06	25.70±0.02	18.46±0.08	3.69±0.10	16.59±0.20	40.92±0.09	15.74±0.21	27.54±0.03
AvgTrans	29.77±0.12	11.36±0.10	20.36±0.19	33.58±0.06	12.69±0.05	27.63±0.11	20.56±0.15	6.96±0.13	16.76±0.09	42.56±0.08	18.63±0.08	28.37±0.03
FuzzyS2S	32.10±0.09	12.14±0.12	23.24±0.09	**35.13±0.12**	**13.26±0.04**	**28.25±0.11**	**23.76±0.08**	**7.28±0.12**	**16.88±0.07**	**42.83±0.11**	**20.26±0.01**	**30.45±0.10**

TABLE IV
RESULTS OF CODE GENERATION TESTS

Model	HS			MTG			GEO			Spider		
	ACC	BLEU	METEOR	ACC	BLEU	METEOR	ACC	BLEU	METEOR	ACC	BLEU	METEOR
CodeT5-Small	3.85±0.12	36.56±0.07	37.34±0.11	2.44±0.01	46.98±0.01	64.93±0.12	7.39±0.03	43.00±0.02	62.42±0.09	3.25±0.04	16.19±0.09	43.19±0.05
CodeT5-Base	3.55±0.06	63.56±0.12	72.14±0.03	2.64±0.05	60.01±0.04	71.57±0.04	7.88±0.08	66.21±0.07	77.57±0.03	5.39±0.02	26.55±0.00	54.08±0.04
CodeT5-Large	4.53±0.09	66.02±0.05	75.67±0.00	3.38±0.11	73.47±0.07	76.97±0.02	8.49±0.09	84.18±0.23	88.50±0.13	5.55±0.01	**26.98±0.01**	**57.49±0.03**
E2S2-T5	4.36±0.05	66.33±0.0	75.21±0.02	3.19±0.08	72.68±0.01	76.40±0.05	8.96±0.04	84.11±0.05	88.01±0.06	8.46±0.08	26.30±0.03	56.63±0.04
Transformer	44.67±0.05	70.82±0.06	70.44±0.03	27.34±0.05	58.70±0.04	63.10±0.05	83.47±0.12	89.65±0.05	89.66±0.07	36.29±0.06	21.48±0.04	46.28±0.08
AvgTrans	45.69±0.17	70.66±0.02	71.32±0.16	28.00±0.12	63.59±0.17	73.61±0.14	85.11±0.06	90.33±0.03	90.52±0.01	35.99±0.12	22.01±0.03	47.01±0.04
FuzzyS2S	**48.89±0.07**	**72.14±0.14**	**76.15±0.08**	**30.30±0.07**	**73.98±0.10**	**76.98±0.04**	**87.33±0.08**	**91.66±0.13**	**92.54±0.04**	**36.35±0.04**	22.33±0.07	48.60±0.07

TABLE V
RESULTS OF FUZZYS2S ABLATION TESTS

Model	EUconst			SAMSum			HS		
	ACC	BLEU	METEOR	R1	R2	RL	ACC	BLEU	METEOR
B+F+G	**55.61±0.11**	**49.64±0.04**	**79.12±0.04**	**35.13±0.10**	**13.26±0.09**	**28.25±0.07**	**48.89±0.01**	**72.14±0.08**	**76.15±0.05**
B+S_1+G	55.31±0.07	46.05±0.07	73.23±0.06	30.91±0.01	7.43±0.11	26.51±0.06	47.22±0.13	71.01±0.09	72.16±0.03
B+S_2+G	55.42±0.05	47.23±0.04	74.11±0.10	32.21±0.08	8.54±0.07	27.23±0.07	47.56±0.09	71.56±0.04	72.96±0.08
B+S_3+G	55.53±0.12	48.07±0.06	75.33±0.12	32.88±0.10	10.90±0.05	27.92±0.10	47.63±0.14	71.87±0.06	73.52±0.04
B+G	55.39±0.06	46.01±0.11	73.20±0.13	30.95±0.06	7.48±0.09	26.54±0.01	47.82±0.05	71.11±0.04	72.15±0.11
B+T	54.65±0.01	45.59±0.07	72.63±0.06	30.16±0.01	6.03±0.05	25.70±0.08	44.67±0.13	70.82±0.11	70.44±0.06

*"B" refers to basic tokenizer, "F" refers to fuzzy tokenizer, "S_1", "S_2" and "S_3" refers to a single coarse, medium and fine tokenizer respectively, "G" refers to the GenFS-Trans module which contains multiple transformers, and "T" refers to a single transformer.

TABLE VI

FUZZY TOKENIZER ABLATION TEST IN TERM OF TRAINING SPEED

Model	EUconst			SAMSum			HS		
	Speed *min* @epoch	CPU% @epoch	GPU% @epoch	Speed *min* @epoch	CPU% @epoch	GPU% @epoch	Speed *min* @epoch	CPU% @epoch	GPU% @epoch
B+F+G	2.11±0.09	3.21±0.07	13.23±0.04	11.90±0.05	4.20±0.18	35.11±0.18	1.51±0.22	2.11±0.08	11.96±0.14
B+S_1+G	1.90±0.18	3.61±0.22	12.11±0.18	11.22±0.06	3.61±0.20	34.21±0.12	1.24±0.04	2.54±0.15	10.23±0.15
B+S_2+G	2.23±0.12	3.70±0.20	12.34±0.13	12.69±0.03	3.58±0.11	36.10±0.20	1.91±0.07	2.22±0.03	11.68±0.01
B+S_3+G	3.01±0.15	3.58±0.04	15.01±0.11	13.54±0.19	3.55±0.14	37.18±0.13	2.70±0.08	2.65±0.14	13.32±0.09

2) Analysis of Summary Generation Tests

The results of summary generation tests are presented in Table III. The results demonstrate that FuzzyS2S outperforms AvgTrans on all datasets, indicating that FuzzyS2S's weighted sum method is better than AvgTrans's averaged sum method. Furthermore, FuzzyS2S exhibits superior performance on the three datasets (SAMSum, XLSum and BillSum) when compared to models of the T5 family, where the value of RL is 5.23 higher on average. Since the input for the summary generation task is an entire article or a report, which constitutes a long sequence, the problem of long-term dependency is significant. One potential solution is to introduce attention mechanisms. The generative rule consequent of FuzzyS2S is a Transformer with a multi-head attention mechanism. Additionally, there is a layer of rule-level soft attention mechanism on top of all the generative rule consequents of FuzzyS2S. This suggests that FuzzyS2S is well-suited for the summary generation task and favorable test results are thus obtained.

3) Analysis of Code Generation Tests

The test results on the code generation dataset are shown in Table IV. The results show that the accuracy of FuzzyS2S is significantly better than that of CodeT5 on all datasets, with an average of 28.69. Compared with classical Transformer, FuzzyS2S also demonstrates a higher accuracy of 2.77 on average. On the HS, MTG, and GEO datasets, the fluency of FuzzyS2S is also better than or close to CodeT5. Since CodeT5 utilizes relative positional encoding, due to the strong correlations between tokens in natural language sequences, relative positional encoding is particularly effective in enhancing fluency when processing natural language. However, code sequences, unlike natural language, have weaker inter-word-element correlations. Thus, CodeT5 method that is based on relative positional coding does not show significant performance improvement in fluency when compared to FuzzyS2S and Transformer.

C. Ablation Test

Table V shows the results of ablation test conducted evaluating the performance with different combinations of FuzzyS2S components. The components include basic tokenizer, fuzzy tokenizer, coarse tokenizer, medium tokenizer, fine tokenizer, GenFS-Trans module and single transformer. The results in the first four rows indicate that the fuzzy tokenizer yields superior model performance when compared to using any single-scale tokenizer (coarse, medium, or fine). Comparing the results of the first row with the fifth row, it is evident that the fuzzy tokenizer contributes to model performance enhancement on almost all the datasets. Furthermore, comparing the second and the third row, it is clear that GenFS-Trans enhances performance across all datasets. This enhancement is also evident in machine translation and summary generation tasks, validating the effectiveness of GenFS-Trans in contributing to FuzzyS2S's overall performance. Further analysis of the ablation tests is given in Part 4 of the *Supplementary Materials*. To assess the impact of tokenization scale on training efficiency, we performed ablation study on the fuzzy tokenizer. As shown in Table VI, the coarse-scale tokenizer yields the shortest training time, followed by medium-scale tokenizer, and the fine-scale tokenizer incurs the highest training cost. This is because fine-scale tokenizer produces more tokens and increases model fitting time. The training time of fuzzy tokenizer is close to that of the medium-scale tokenizer, indicating that a balance between token size and computational cost is achieved through multi-scale integration.

D. Convergence Analysis

The convergence of FuzzyS2S when trained on EUconst, SAMSum, and HS datasets are illustrated in Fig. 4. From the figure, it can be observed that the loss of FuzzyS2S tends to converge after 20, 30 and 20 epochs of training on the three datasets, respectively. The test results demonstrate that FuzzyS2S exhibits relatively good stability and convergence.

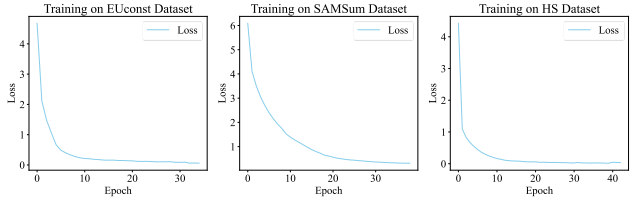


Fig.4. Convergence Analysis of FuzzyS2S on EUconst, SAMSum, HS Datasets

E. Interpretability Analysis

The interpretability of FuzzyS2S was analyzed and demonstrated through a case study as shown in Fig. 5, where English sentences were translated into French from the Tatoeba dataset. In the test, we configured the GenFS-Trans module in FuzzyS2S to contain the three rules below:

$$\begin{aligned}&\text{IF } \boldsymbol{s}_x \text{ is } \textit{Long Sentence}, \text{ THEN } \boldsymbol{s}_y = TF^{Long}(\boldsymbol{s}_x);\\ &\text{IF } \boldsymbol{s}_x \text{ is } \textit{Middle Sentence}, \text{ THEN } \boldsymbol{s}_y = TF^{Middle}(\boldsymbol{s}_x);\\ &\text{IF } \boldsymbol{s}_x \text{ is } \textit{Short Sentence}, \text{ THEN } \boldsymbol{s}_y = TF^{Low}(\boldsymbol{s}_x);\end{aligned} \tag{25}$$

The input English sentence in Fig. 5 has a length of 16, and its length and token frequency distribution characteristics are more similar to the description of the second rule. The actual similarity calculation indicates that the input sentence is more similar to the delegate in the *Middle Sentence* fuzzy set. Consequently, the result obtained through the GFRCM mechanism is consistent with the output of the *Middle Sentence* fuzzy rule. This example demonstrates that when the features of the input sentence align with the fuzzy term

descriptions of the generative rule antecedent, the final output closely matches the output of the corresponding rule. This shows that the generative process of FuzzyS2S is semantically interpretable, which verifies the semantic interpretability of GenFS.

To evaluate the term-based interpretability of GenFS, we randomly selected 100 sentences from each of the four datasets – WMT14, Tatoeba, EUconst, and Ubuntu, and labeled them as "Long", "Medium", or "Short". We compared GenFS classification with human expert judgment. As shown in Fig. 6, the agreement rates are 97%, 98%, 94%, and 97%, respectively, indicating that GenFS's term-based predictions are closely well-aligned with human assessment, thereby demonstrating the model's interpretability.

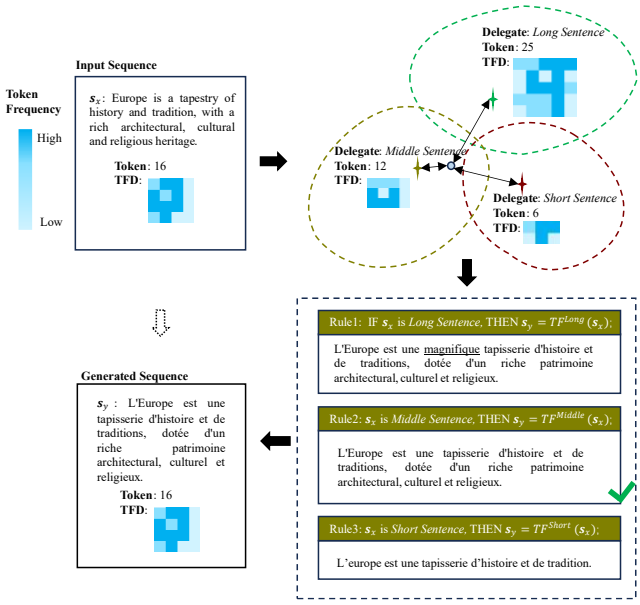


Fig. 5. An example of interpretability analysis: long, middle and short English sentences are translated into French in FuzzyS2S. The length of the delegate of *Long Sequence* fuzzy set is 25, the length of the delegate of *Middle Sequence* fuzzy set is 12, the length of the delegate of *Short Sequence* fuzzy set is 6, and TFD refers to Token Frequency Distribution.

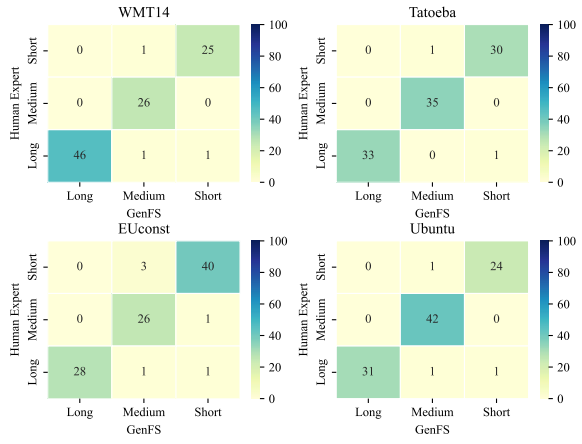


Fig. 6. Sentence length term-based consistency comparison between the GenFS model and human experts across four datasets (WMT14, Tatoeba, EUconst and Ubuntu).

F. Further Analysis

In FuzzyS2S, fuzzy tokenizer is capable of optimizing the token frequency distribution through multi-scale tokenization. The capability is analyzed in Part 5 of the *Supplementary Materials*.

Besides, number of rules is a key hyperparameter in FuzzyS2S. The number of rules increases linearly with the model's spatial complexity and GPU memory usage. While incorporating more rules can improve the capability of FuzzyS2S, the risk of overfitting increases and performance degrades. Thus, there exists a trade-off between performance and the number of rules. To this end, we conducted parameter sensitivity analysis on the number of rules which is discussed in Part 6 of the *Supplementary Materials*.

VI. Conclusion

The learning process of current generative models is typically data-driven. It lacks knowledge-driven mechanisms and the modeling process is like a black box. To address these issues, we propose a novel generative modeling framework based on fuzzy systems, named GenFS, which is capable of handling complex generative modeling tasks effectively. Furthermore, GenFS incorporates the classical fuzzy system's dual-driven learning mechanism of both data and knowledge to offer semantical interpretability. For sequence-to-sequence generative learning tasks, we propose a novel end-to-end generative model based on GenFS framework, named FuzzyS2S, which is effective for machine translation, summary generation, and code generation. Our test results demonstrate that FuzzyS2S significantly outperforms the classical Transformer and surpasses the state-of-the-art T5 family models.

Despite its outstanding performance, GenFS has several limitations. The consequents of GenFS lack interpretability. It is limited to text generation and does not support multimodal generative tasks. Future work will focus on these two issues. First, the interpretability of GenFS consequents will be enhanced by introducing intrinsically interpretable modeling approaches, and by mapping model parameters – decomposed through a Mixture-of-Experts framework – to yield interpretable terms. Second, research will be conducted to unify the modeling of multimodal input and output. To achieve this goal, we will convert multimodal data into unified term-based representations, perform inference over the intermediate representations, and subsequently generate outputs across different modalities.

References

[1] B. Min *et al.*, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[2] W. X. Zhao *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[3] H. Touvron *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[4] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, "Generative AI," *Bus Inf Syst Eng*, vol. 66, no. 1, pp. 111–126, Feb. 2024, doi: 10.1007/s12599-023-00834-7.

[6] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[7] D. Dai *et al.*, "Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers," *arXiv preprint arXiv:2212.10559*, 2022.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[9] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 8748–8763.

[10] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[11] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. Ellis, "Mulan: A joint embedding of music audio and natural language," *arXiv preprint arXiv:2208.12415*, 2022.

[12] Wang Z., Gui L., Negrea J., and Veitch V., "Concept Algebra for (Score-Based) Text-Controlled Generative Models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 35331–35349, Dec. 2023.

[13] B. C. Challagundla and C. Peddavenkatagari, "Neural Sequence-to-Sequence Modeling with Attention by Leveraging Deep Learning Architectures for Enhanced Contextual Understanding in Abstractive Text Summarization," Apr. 08, 2024, *arXiv*: arXiv:2404.08685. doi: 10.48550/arXiv.2404.08685.

[14] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat Mach Intell*, vol. 1, no. 5, pp. 206–215, May 2019, doi: 10.1038/s42256-019-0048-x.

[15] S. Guillaume, "Designing fuzzy inference systems from data: An interpretability-oriented review," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 3, pp. 426–443, Jun. 2001, doi: 10.1109/91.928739.

[16] Y. Zhang, H. Ishibuchi, and S. Wang, "Deep Takagi–Sugeno–Kang Fuzzy Classifier with Shared Linguistic Fuzzy Rules," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1535–1549, Jun. 2018, doi: 10.1109/TFUZZ.2017.2729507.

[17] T. Zhou, F.-L. Chung, and S. Wang, "Deep TSK Fuzzy Classifier with Stacked Generalization and Triplely Concise Interpretability Guarantee for Large Data," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1207–1221, Oct. 2017, doi: 10.1109/TFUZZ.2016.2604003.

[18] V. Singh, R. Dev, N. K. Dhar, P. Agrawal, and N. K. Verma, "Adaptive Type-2 Fuzzy Approach for Filtering Salt and Pepper Noise in Grayscale Images," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3170–3176, Oct. 2018, doi: 10.1109/TFUZZ.2018.2805289.

[19] X. Ma *et al.*, "Deep Image Feature Learning with Fuzzy Rules," *IEEE Trans. Emerg. Top. Comput. Intell.*, pp. 1–14, 2023, doi: 10.1109/TETCI.2023.3259447.

[20] X. Tian *et al.*, "Deep Multi-View Feature Learning for EEG-Based Epileptic Seizure Detection," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 10, pp. 1962–1972, Oct. 2019, doi: 10.1109/TNSRE.2019.2940485.

[21] C. Pozna and R.-E. Precup, "Aspects Concerning the Observation Process Modelling in the Framework of Cognition Processes," *Acta Polytechnica Hungarica*, vol. 9, no. 1, pp. 203–223, 2012.

[22] Z. Deng, Y. Jiang, H. Ishibuchi, K.-S. Choi, and S. Wang, "Enhanced Knowledge-Leverage-Based TSK Fuzzy System Modeling for Inductive Transfer Learning," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 1, pp. 1–21, Jan. 2017, doi: 10.1145/2903725.

[23] P. Xu, Z. Deng, J. Wang, Q. Zhang, K.-S. Choi, and S. Wang, "Transfer Representation Learning with TSK Fuzzy System," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 3, pp. 649–663, Mar. 2021, doi: 10.1109/TFUZZ.2019.2958299.

[24] J. Deshmukh, A. K. M, and S. Sengupta, "A Sequence Modeling Approach for Structured Data Extraction from Unstructured Text," in *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, L. Espinosa-Anke, T. Declerck, D. Gromann, J. Camacho-Collados, and M. T. Pilehvar, Eds., Macau, China: Association for Computational Linguistics, Aug. 2019, pp. 57–66.

[25] E. Tulchinskii *et al.*, "Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., Curran Associates, Inc., 2023, pp. 39257–39276.

[26] J. Zhai, S. Zhang, J. Chen, and Q. He, "Autoencoder and Its Various Variants," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan: IEEE, Oct. 2018, pp. 415–419. doi: 10.1109/SMC.2018.00080.

[27] I. Goodfellow *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[28] J. Song, C. Meng, and S. Ermon, "Denoising Diffusion Implicit Models," in *9th International Conference on Learning Representations*, OpenReview.net, May 2021.

[29] K. Bai *et al.*, "A data-knowledge-driven interval type-2 fuzzy neural network with interpretability and self-adaptive structure," *Information Sciences*, vol. 660, p. 120133, Mar. 2024, doi: 10.1016/j.ins.2024.120133.

[30] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2014.

[31] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[32] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.

[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[34] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," Apr. 03, 2021, *arXiv*: arXiv:2003.05991.

[35] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, p. 140:5485-140:5551, Jan. 2020.

[36] J. Achiam *et al.*, "GPT-4 Technical Report," Mar. 04, 2024, *arXiv*: arXiv:2303.08774.

[37] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, vol. 11141, 2018, pp. 270–279.

[38] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012, pp. 17–36.

[39] G. Mesnil *et al.*, "Unsupervised and transfer learning challenge: a deep learning approach," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 97–110.

[40] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, May 2019, doi: 10.1038/s42256-019-0048-x.

[41] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, vol. 63, no. 1, pp. 68–77, Dec. 2019, doi: 10.1145/3359786.

[42] Q. Zhong, L. Ding, J. Liu, B. Du, and D. Tao, "E2S2: Encoding-Enhanced Sequence-to-Sequence Pretraining for Language Understanding and Generation," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 12, pp. 8037–8050, Dec. 2024, doi: 10.1109/TKDE.2023.3341917.

[43] P. Hall, "On Representatives of Subsets," in *Classic Papers in Combinatorics*, 1987, pp. 58–62. doi: 10.1007/978-0-8176-4842-8_4.

[44] M. Hall Jr, "Distinct representatives of subsets," *Bulletin of the American Mathematical Society*, vol. 54, no. 10, pp. 922–926, 1948.

[45] R.-E. Precup, C.-A. Bojan-Dragos, R.-C. Roman, and E. M. Petriu, "Evolving Fuzzy Models of Shape Memory Alloy Wire Actuators," *Romanian Journal of Information Science and Technology*, vol. 24, no. 4, pp. 353–365, 2021.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[47] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam Optimization Algorithm for Wide and Deep Neural Network," *Kno. Eng. Da. Sc.*, vol. 2, no. 1, p. 41, Jun. 2019, doi: 10.17977/um018v2i12019p41-46.

[48] D. M. W. Powers, "Applications and explanations of Zipf's law," in *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning -*

NeMLaP3/CoNLL '98, Sydney, Australia: Association for Computational Linguistics, 1998, p. 151. doi: 10.3115/1603899.1603924.

[49] Z. Shao, M. Huang, J. Wen, W. Xu, and X. Zhu, "Long and Diverse Text Generation with Planning-based Hierarchical Variational Model," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3255–3266. doi: 10.18653/v1/D19-1321.

[50] M. Cheng, J. Yi, P.-Y. Chen, H. Zhang, and C.-J. Hsieh, "Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples," *AAAI*, vol. 34, no. 04, pp. 3601–3608, Apr. 2020, doi: 10.1609/aaai.v34i04.5767.

[51] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds., Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. doi: 10.18653/v1/P16-1162.

[52] S. Medasani, J. Kim, and R. Krishnapuram, "An overview of membership function generation techniques for pattern recognition," *International Journal of Approximate Reasoning*, vol. 19, no. 3–4, pp. 391–417, Oct. 1998, doi: 10.1016/S0888-613X(98)10017-8.

[53] Hao Ying, "General SISO Takagi-Sugeno fuzzy systems with linear rule consequent are universal approximators," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 4, pp. 582–587, Nov. 1998, doi: 10.1109/91.728456.

[54] X.-J. Zeng and M. G. Singh, "Approximation accuracy analysis of fuzzy systems as function approximators," *IEEE Transactions on fuzzy systems*, vol. 4, no. 1, pp. 44–63, 1996.

[55] B. Kosko, "Fuzzy systems as universal approximators," *IEEE transactions on computers*, vol. 43, no. 11, pp. 1329–1333, 1994.

[56] L.-X. Wang, "Fuzzy systems are universal approximators," in *[1992 proceedings] IEEE international conference on fuzzy systems*, 1992, pp. 1163–1170.

[57] W. A. Qader, M. M. Ameen, and B. I. Ahmed, "An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges," in *2019 International Engineering Conference (IEC)*, Erbil, Iraq: IEEE, Jun. 2019, pp. 200–204. doi: 10.1109/IEC47844.2019.8950616.

[58] C.-F. Tsai, "Bag-of-Words Representation in Image Annotation: A Review," *ISRN Artificial Intelligence*, vol. 2012, pp. 1–19, Nov. 2012, doi: 10.5402/2012/376804.

[59] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *Int. J. Mach. Learn. & Cyber.*, vol. 1, no. 1, pp. 43–52, Dec. 2010, doi: 10.1007/s13042-010-0001-0.

[60] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2–3, pp. 191–203, Jan. 1984, doi: 10.1016/0098-3004(84)90020-7.

[61] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, Nov. 1980, doi: 10.1016/0146-664X(80)90054-4.

[62] R. Suwanda, Z. Syahputra, and E. M. Zamzami, "Analysis of Euclidean Distance and Manhattan Distance in the K-Means Algorithm for Variations Number of Centroid K," *J. Phys.: Conf. Ser.*, vol. 1566, no. 1, p. 012058, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012058.

[63] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," in *2016 4th International Conference on Cyber and IT Service Management*, Bandung, Indonesia: IEEE, Apr. 2016, pp. 1–6. doi: 10.1109/CITSM.2016.7577578.

[64] D. Gunawan, C. A. Sembiring, and M. A. Budiman, "The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents," *J. Phys.: Conf. Ser.*, vol. 978, p. 012120, Mar. 2018, doi: 10.1088/1742-6596/978/1/012120.

[65] O. Bojar *et al.*, "Findings of the 2014 Workshop on Statistical Machine Translation," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA: Association for Computational Linguistics, Jun. 2014, pp. 12–58. doi: 10.3115/v1/W14-3302.

[66] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 569–575, Mar. 2010, doi: 10.1109/TPAMI.2009.187.

[67] J. Tiedemann, "The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT," in *Proceedings of the Fifth Conference on Machine Translation*, Online: Association for Computational Linguistics, Nov. 2020, pp. 1174–1182.

[68] J. Tiedemann, "Parallel Data, Tools and Interfaces in OPUS," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 2214–2218.

[69] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.

[70] K. M. Hermann *et al.*, "Teaching machines to read and comprehend," *Advances in neural information processing systems*, vol. 28, 2015.

[71] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 70–79. doi: 10.18653/v1/D19-5409.

[72] T. Hasan *et al.*, "XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online: Association for Computational Linguistics, Aug. 2021, pp. 4693–4703. doi: 10.18653/v1/2021.findings-acl.413.

[73] A. Kornilova and V. Eidelman, "BillSum: A Corpus for Automatic Summarization of US Legislation," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 48–56. doi: 10.18653/v1/D19-5406.

[74] W. Ling *et al.*, "Latent predictor networks for code generation," *arXiv preprint arXiv:1603.06744*, 2016.

[75] T. Yu *et al.*, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," *arXiv preprint arXiv:1809.08887*, 2018.

[76] Y. Wang, W. Wang, S. Joty, and S. C. Hoi, "Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation," *arXiv preprint arXiv:2109.00859*, 2021.

[77] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. doi: 10.3115/1073083.1073135.

[78] A. Lavie and A. Agarwal, "Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments," in *Proceedings of the Second Workshop on Statistical Machine Translation - StatMT '07*, Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 228–231. doi: 10.3115/1626355.1626389.

[79] P. Dufter, M. Schmitt, and H. Schütze, "Position Information in Transformers: An Overview," *Computational Linguistics*, vol. 48, no. 3, pp. 733–763, Sep. 2022, doi: 10.1162/coli_a_00445.

[80] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[81] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021, doi: 10.1016/j.neucom.2021.03.091.

[82] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An Attentive Survey of Attention Models," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 5, pp. 1–32, Oct. 2021, doi: 10.1145/3465055.

[83] K. Xu *et al.*, "Show, attend and tell: neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, in ICML'15. Lille, France: JMLR.org, Jul. 2015, pp. 2048–2057.

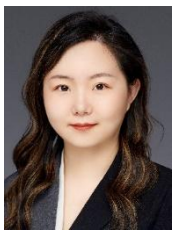
Hailong Yang is currently working toward the Ph.D. degree in software engineering with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China.
His research interests include generative fuzzy system, fuzzy modeling, multi-agent system and their applications.

Zhaohong Deng (M'12-SM'14) received the B.S. degree in physics from Fuyang Normal College, Fuyang, China, in 2002, and the Ph.D. degree in information technology and engineering from Jiangnan University, Wuxi, China, in 2008.

He is currently a Professor with the School of Artificial Intelligence and Computer Science, Jiangnan University. He has visited the University of California-Davis and the Hong Kong Polytechnic University for more than two years. His current research interests include interpretable intelligence, uncertainty in artificial intelligence and their applications. He has authored or coauthored more than 100 research papers in international/ national journals.
Dr. Deng has served as an Associate Editor or Guest Editor of several international Journals, such as IEEE Trans. Emerging Topics in Computational Intelligence, Neurocomputing, and so on.

Wei Zhang is currently working toward the Ph.D. degree in light industry information technology with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China.
His research interests include computational intelligence, machine learning, interpretable artificial intelligence, fuzzy modeling and their applications.

Zhuangzhuang Zhao is currently working toward the Ph.D. degree in software engineering with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China.
His research interests include fuzzy computing, machine learning, interpretable artificial intelligence, fuzzy modeling.



Guanjin Wang received a joint Ph.D. degree in software engineering from the University of Technology Sydney, NSW, Australia, and The Hong Kong Polytechnic University, Hong Kong. She is currently a senior lecturer in the School of Information Technology at Murdoch University, Perth, Australia. Her recent research focuses on developing cutting-edge methods for learning from complex and imperfect data and learning environments, as well as achieving model explainability. She has published high-quality research papers in journals such as IEEE TCYB, IEEE TFS, IEEE SMC, and IEEE J-BHI.

Kup-Sze Choi (M'97-SM'23) received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong in 2004.
He is a former Professor at the School of Nursing, Hong Kong Polytechnic University, Hong Kong, and the Director of the Centre for Smart Health. His research interests include virtual reality, artificial intelligence, and their applications in medicine and healthcare.