ICIC Express Letters
Volume **17**, Number **1**, January **2023**

ICIC International ©2023 ISSN 1881-803X
pp. **25–30**

REINFORCEMENT LEARNING APPROACH USING FUZZY-ROUGH SET THEORY FOR MACHINE TRANSLATION

Omnia Yahia, Yasser Fouad Hassan and Ashraf Elsayed

Department of Mathematics and Computer Science
Faculty of Science
Alexandria University
22 El-Guish Road, El-Shatby, Alexandria 21526, Egypt
omniayahia2411@gmail.com; y.fouad@alexu.edu.eg; aselsayed@zewailcity.edu.eg

Received October 2021; accepted January 2022

Abstract. *Machine translation cares about the translation of the word in the context of the sentence not the translation of the word itself to get a meaningful translation. In this paper, we introduce the reinforcement learning (RL) approach to translate English sentences into Arabic sentences as RL aims to teach the agent how to choose the action according to the state in order to maximize the final reward. We have integrated the fuzzy rough set (FRS) theory with RL, a rough set approach to dealing with vagueness and uncertainty in decision making so we can extract translation rules for each word, and fuzzy logic to generate a fuzzy reward signal for the decisional attribute in the decision system. To verify our method, the proposed model was applied to a set of manually translated sentences from the United Nations website to compare our results with these manual translations and our proposed model has achieved significant improvement in the translation.*

Keywords: Machine translation, Fuzzy rough set, Reinforcement learning, Q-learning

1. **Introduction.** Machine translation aims at obtaining high-quality translation for a text from one language to another within seconds and without the need for humans. However, the fact is that accurate translation requires background knowledge in order to resolve ambiguity and establish the content of the sentence. Some previous studies in machine translation have relied on the RL approach because RL has proven its efficiency in many applications that are concerned with learning the agent to take the most proper action in each state without the need for a supervisor, by rewarding the agent for its action with positive or negative reward and the agent seeks to maximize the final reward. Also, RL incorporates external memory to store the decisions that have the highest reward. Integration of RL with machine translation has been shown that RL is an effective approach for improving the performance of the neural machine translation (NMT) system [1-3]. Q-learning is one of the most popular RL algorithms and we have chosen it for our study because it has had successful results in many previous studies [4,5]. In this study, we seek to improve previous results in RL and machine translation studies, so we integrated the fuzzy rough set (FRS) theory with RL to reduce the uncertainty in decision making and to get the reward value as a fuzzy membership. FRS has proven its efficiency in reducing uncertainty in many previous studies [6-8]. In this study, we will show the advantages of combining RL, fuzzy rough set (FRS), and machine translation. Comparing this study with previous studies, we found that this new algorithm achieved a better result in machine translation, especially with sentences that contain words that have different meanings. The rest of this paper is structured as follows. We present the FRS approach, and RL in Section 2. Section 3 talks about the proposed methodology. Section 4 applies our

DOI: 10.24507/icicel.17.01.25

proposed method to English sentences and interprets the result and discussions. Finally, the conclusion part of this proposed methodology is presented in Section 5.

2. Background.

2.1. Reinforcement learning.

RL is defined as a machine learning method that enables the agent to learn how to choose the most proper action between a set of possible actions for the current state through trial-and-error experience. In the exploration episodes, the agent will choose random action from the possible actions, the environment takes the agent's current state and action as input, and returns as output the agent's reward and its next state, this reward is a short-term reward but the agent cares about the final reward, the agent through these exploration episodes learns the strategy that should follow to accomplish its tasks, and this strategy is called the policy. After the exploration, the agent starts the exploitation to learn from its experience.

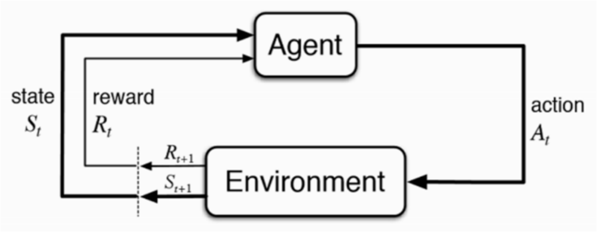

Figure 1. Reinforcement learning

2.2. Fuzzy rough set.

FRS is a generalization of a rough set, derived from the approximation of a fuzzy set in a crisp approximation space. This corresponds to the case where the values of conditional attribute are crisp and the decision attribute values are fuzzy. Rough sets can be expressed by a fuzzy membership function to represent the negative, boundary, and positive regions. In this model, the elements belonging to lower approximation or positive region have a membership value of one, those belonging to boundary region have a membership value of 0.5 and those elements belonging to upper approximation or negative region have a membership value of 0. Fuzziness is integrated into rough sets which use fuzzy membership values to qualify levels of roughness in boundary region.

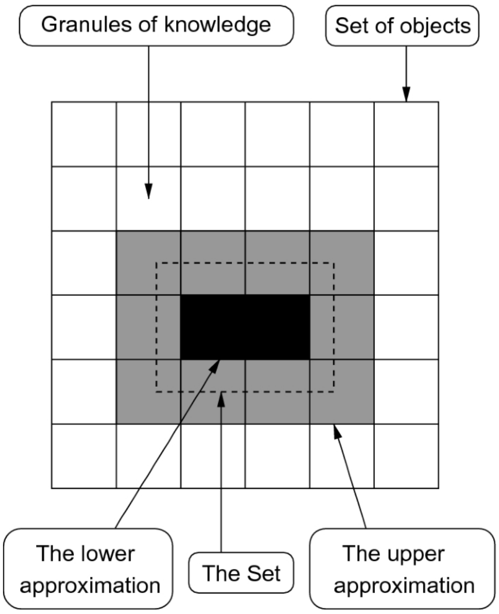

Figure 2. Rough set theory

Suppose, R is an equivalence relation, which is imposed on the universe U. The equivalence class is expressed as fuzzy sets $F = \{F_1, F_2, \ldots, F_H\}$, when the classes to which the elements attribute are ambiguous. F_i is a fuzzy set, $i \in \{1, 2, \ldots, H\}$. The fuzzy lower and upper approximations are given as

$$\mu_{\underline{p}x}(F_i) = \inf_x \max\{1 - \mu_{F_i}(x), \mu_x(x)\}$$

$$\mu_{\bar{p}x}(F_i) = \sup_x \min\{\mu_{F_i}(x), \mu_x(x)\}$$

FRS is an integration of fuzzy set advantages and rough set advantages, so it has been used to solve practical problems such as data mining and approximate reasoning and it has received many interesting results.

The goal of our work is to combine the advantages of the RL algorithm with the advantages of the FRS approach.

3. Proposed Methodology.

3.1. Preprocessing.

We should preprocess the text before starting to translate it, we will use NLTK for preparing the text. NLTK provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing, libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Preprocessing steps:

- Split the text into sentences and split the sentences into tokens.
- Remove punctuation.
- Filter out stop words (the, at...).
- Classify the words into their parts of speech.
- Convert all words to one case "lowercase".

3.2. Decision system.

We will use a dataset of English sentences translated into Arabic sentences to create the decision system which consists of conditional attributes and decision attributes. The conditional attributes are the word "i", two words before this word "$i-1$, $i-2$", two words after this word "$i+1$, $i+2$", type of this word. The decision attribute value will be the translation of the word "i" as shown below in Table 1.

TABLE 1. Sample from the decision system

Word(i)	$i-1$	$i-2$	$i+1$	$i+2$	Type	Translation
bats	thousands		colonized	ruins	noun	الخفافيش
bats			feed	insects	noun	الخفافيش
bats			feed	fruit	noun	الخفافيش
bats	dogs	cats	carry	rabies	noun	الخفافيش
bats	clubbed	baseball			noun	مضارب
bats			death	baseball	noun	مضارب
bats	baseball	couple	sports	bag	noun	مضارب
bats	cricket	apple	tennis	rackets	noun	مضارب
bat	trying		flies	away	verb	يحاول
batting	Who's		first	food	verb	يتناول

We use ROSETTA toolkit [9] to analyze the decision system and generate if-then rules.

Rules:

i(bats) AND $i+2$(vibration) AND type(noun)⇒translation(الخفافيش)
i(bats) AND $i-1$(foxes) AND $i+2$(rabies) AND type(noun)⇒translation(الخفافيش)
i(bats) AND $i-1$(dusk) AND $i+2$(fly) AND type(noun)⇒translation(الخفافيش)
i(bats) AND $i+2$(baseball) AND type(noun)⇒translation(مضارب)

i(bats) AND $i-1$(cricket) AND $i+4$(rackets) AND type(noun)⇒translation(مضارب)

Each word in the language has many different meanings, so when the agent starts to select random action "translation" from the available actions, we will check the decision attribute for this state, if the selected action is the decision attribute value, so the fuzzy value for this action will be 1, otherwise 0. We will use this fuzzy membership as a reward for the selected action in the Q-learning equation.

3.3. **Reinforcement learning.** The agent's task is to translate an English sentence into an Arabic sentence, the agent will consider the first word w_i in this sentence the first state.

Initialize available actions for each state so the agent can select an action randomly from it, for example,

"bats" its available actions: خفافيش, مضارب, ...

"fine" its available actions, if it is a noun are غرامة, نهاية, . . . , and if it is adjective are رفيع, رائع, جميل, ...

$$Q(\text{state,action}) \leftarrow (1-\alpha)Q(\text{state,action}) + \alpha(\text{reward} + \gamma \max Q(\text{next state,all actions})) \tag{1}$$

where α $(0 \leq \alpha \leq 1)$ is the learning rate and γ $(0 \leq \gamma \leq 1)$ is the discount factor.

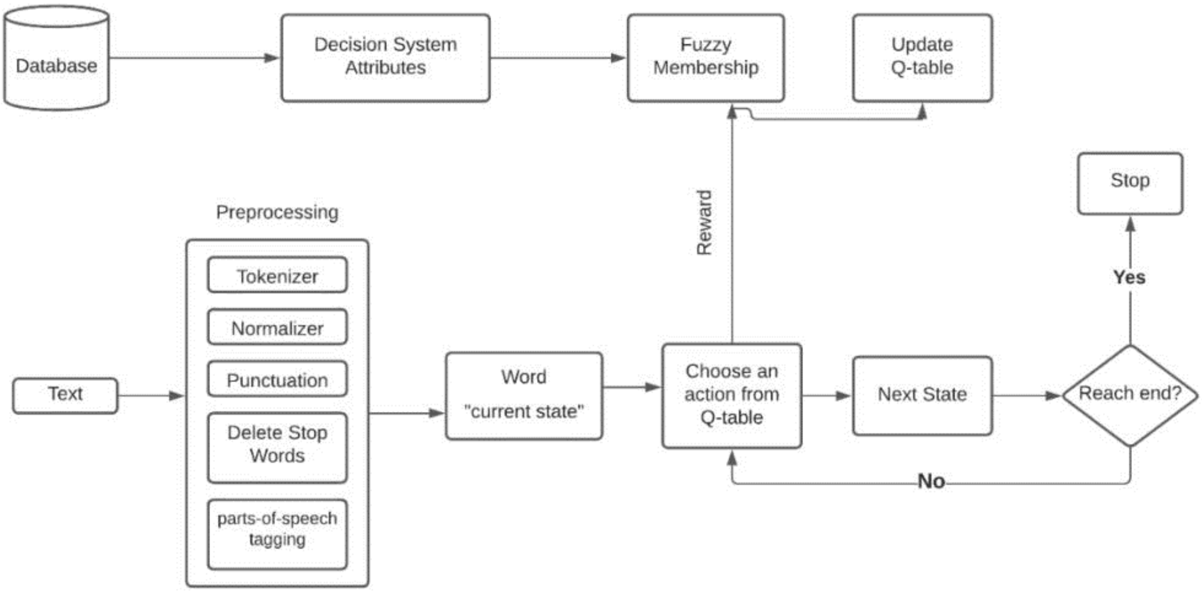

FIGURE 3. Flowchart of the proposed approach

4. **Experiments.** To verify the proposed algorithm, we will test this algorithm on some manually translated English to compare the manual translation with our proposed algorithm.

The proposed algorithm executed was on Windows 10 and coded in python 3.9.1, using Pyharm IDE.

Initialize the discount factor γ to 0.7 and the learning rate α to 0.9.

We will use a dataset from the United Nations website to create the decision system, the conditional attributes are the word "i", two words before this word "$i-1$, $i-2$", two words after this word "$i+1$, $i+2$", type of this word and the decision attribute is the translation, as shown in Table 1. We will use the ROSETTA toolkit to analyze the decision system and generate if-then rules and generate fuzzy membership for the decision attribute.

We will start preprocessing the text to apply our proposed algorithm to it, and we will use NLTK packages to prepare the text (sent_tokenize, word_tokenize, stopwords, and pos_tag). Divide the text into sentences, divide each sentence into words and remove the

Proposed Algorithm: Q-learning

```
1:  initialize Q-table to zeros
2:  for $i = 1$ to iterations do
3:      for episode = 1 to sentence length do
4:      observe the current state $s_t$
5:      choose random action $a_t$ from Q-table
6:      perform the action $a_t$
7:      receive a reward from the decision system
8:      calculate the $Q(s_t, a_t)$ value according to Equation (1)
9:      update the Q-table
10:     end for
11: end for
12: observe the current state $s_t$
13: (the exploitation) choose action with maximum Q-value in the Q-table from the
    available actions
14: do steps from 6 to 9
```

punctuation and filter out the stop words and convert the text to lowercase, and finally classify the words into their parts of speech.

Initialize the Q-table $(n*m)$, where n is the number of rows that represent the states and m is the number of columns that represent the Arabic translations and we will initialize its values to zeros.

We have set up the environment so the agent can start exploring it, the agent will explore the environment through 1000 iterations, and in each iteration the agent tries to learn how to translate the sentence. The first word in the sentence is the initial state, then the agent will choose random action from the available actions for this word, and then the agent will receive from the environment an immediate reward for its action which is 0 or 1 according to the decision system. Then we will calculate the Q-value of this action in this state according to Equation (1) and then update the Q-table.

$$\text{Q-matrix} = \begin{bmatrix} Q(s_1, a_1) & Q(s_1, a_2) & \cdots & Q(s_1, a_m) \\ Q(s_2, a_1) & Q(s_2, a_2) & \cdots & Q(s_2, a_m) \\ \vdots & \vdots & \ddots & \vdots \\ Q(s_n, a_1) & Q(s_n, a_2) & \cdots & Q(s_n, a_m) \end{bmatrix}$$

The agent will move to the next state "next word" and continue exploring the environment until finishing the iterations. Then the agent starts to exploit its knowledge and choose the action with the maximum value in the Q-table.

After the agent has learned through the error and trial, we evaluated the results by translating the same sentences manually and we found that the proposed algorithm has good results, for example,

Examples of the algorithm translation:

Thousands of **bats** had colonized the ruins.	استعمرت آلاف **الخفافيش** الأنقاض .
Bats feed on insects and fruit.	تتغذى **الخفافيش** على الحشرات والفواكه.
Thugs clubbed him with baseball **bats**.	ضربه البلطجية **بمضارب** البيسبول .
Baseball **bats**.	**مضارب** بيسبول
Nobody likes to live in the **desert**.	لا أحد يحب العيش في **الصحراء**.
Vast areas of land have become **desert**.	أصبحت مساحات شاسعة من الأراضي **صحراوية**.
The heat in the **desert** was extreme.	كانت الحرارة في **الصحراء** شديدة.
The villages had been **deserted**.	كانت القرى **مهجورة**
All teachers are **fine** ladies.	جميع المعلمات سيدات **رائعات**.

5. **Conclusion.** In this paper, we have combined RL and FRS to create a new algorithm and we checked its efficiency in machine translation. RL teaches the agent how to choose the most proper action from the available actions according to its current state to maximize the final reward to get the most accurate decision and as discussed before, the core in RL is the reward, so we used in this study the FRS to reduce uncertainty data for RL and to get the reward as a fuzzy value. In this algorithm firstly we clear our data using RST to remove any redundant data. Then we extract if-then rules, so we can get the reward value. After that, we apply the RL approach and let the agent explore the environment. The combination between RL and FRS leads to a useful algorithm and we checked its efficiency in machine translation and it showed accurate translation and good results.

REFERENCES

[1] S. Kiegeland and J. Kreutzer, Revisiting the weaknesses of reinforcement learning for neural machine translation, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, https://doi.org/10.48550/arXiv.2106.08942, 2021.

[2] L. Wu, F. Tian, T. Qin, J. Lai and T. Liu, A study of reinforcement learning for neural machine translation, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp.3612-3621, doi: 10.18653/v1/D18-1397, 2018.

[3] T. Kin, J. Kreutzer and S. Riezler, A reinforcement learning approach to interactive-predictive neural machine translations, *arXiv.org*, arXiv: 1805.01553, 2018.

[4] M. Gheisarnejad, M. Sharifzadeh, M. Khooban and K. Al-Haddad, Adaptive fuzzy Q-learning control design and application to grid-tied nine-level Packed E-Cell (PEC9) inverter, *IEEE Transactions on Industrial Electronics*, vol.70, no.1, pp.1071-1076, 2022.

[5] V. Ajabshir, M. Guzel and E. Bostanci, A low-cost Q-learning-based approach to handle continuous space problems for decentralized multi-agent robot navigation in cluttered environments, *IEEE Access*, vol.10, pp.35287-35301, 2022.

[6] X. Yang, Y. Li, D. Liu and T. Li, Hierarchical fuzzy rough approximations with three-way multigranularity learning, *IEEE Transactions on Fuzzy Systems*, vol.30, no.9, pp.3486-3500, 2022.

[7] A. Kumar and P. S. V. S. Sai, Scalable fuzzy rough set reduct computation using fuzzy min-max neural network preprocessing, *IEEE Transactions on Fuzzy Systems*, vol.28, no.5, pp.953-964, 2020.

[8] K. Zhang, J. Zhan and W-Z. Wu, On multicriteria decision-making method based on a fuzzy rough set model with fuzzy α-neighborhoods, *IEEE Transactions on Fuzzy Systems*, vol.29, no.9, pp.2491-2505, 2020.

[9] *http://bioinf.icm.uu.se/rosetta/*, 2009.