Symptemist
Name Entity Recognition

Abstract

The field text analysis have concentrated on medical issues that address the diagnoses problem to machine learning. The medical documents can easily annotated only if the corpus does not consist of millions of word. In such case word recognition as entity can be solved using discriminative models and propose new technologies such as reinforcement learning that can reduce the error of annotated medical words in documents. This project demonstrates how name entity recognition can be enhanced to the reinforcement learning using two algorithms the actor critic and deep deterministic policy gradient. The measurements is the precision and the reward of the agent for each time that discovers an entity that belongs to a symptom of a disease.

1. Introduction

The paper has as a main way to identify words that describe symptoms of a disease. Documents are stored in clinics as reports that describe the state of a patient which are hard to find among the other diseases and the diagnosis usually takes a lot of time. In addition, the continuous change of the disease might give different symptoms that depend on the stage of the disease. Artificial intelligence and machine learning are two fields that can make a diagnosis from written documents or electronic reports. The machine learning represents a statistical report that extracts the required information about the disease and classifies the symptoms. Another new field, reinforcement learning, can understand the symptoms. It represents a technology that until now has been used to solve environmental problems such as self-driving cars. However, we can also solve medical problems and gain all the information that is related to the disease with almost the same accuracy as machine learning.

Name entity recognition is a text analysis that can label each word. Until now, the ner has been used to delete specific text from a huge corpus that, in previous years, was a time-consuming problem. It is a slightly different approach than text classification, which handles document and sentence level classification tasks. In the current approach, each word is handled independently.

The initial reinforcement of learning approaches such as the q-learning has already solved game theoretical problems. It is one approach that makes use of the Bellman equation to navigate through the environment. In the current project, as the research around reinforcement learning is at the peak of technology , a different approach is presented that is more efficient and makes use of neural networks. However, it requires a lot of computational power.

The remainder of this paper is organized as follow, in section 2 related work we discuss related work on the of text analysis. Section 4, 1 explains the machie learning and artificial intelligence algorithms and the tuned parameters. In section 7 dataset provides information about the corpus. In section 8 results we provide our experimental results. Finally, in section 9 conclusion, we conclude the software outcome and provide suggestions for future implementation.

2. Related Work

For the Medical dataset that consists of entities name, ID, location, date, age, other, the conditional random fields had a precision of 0.74. The paper describes how to extract dictionary-based features and that the word suffix 3,2 and the removal of punctuation and the capital letters to lower can improve the classifier performance (A.Agrafiots, 2019).

3. Feature extraction

The dataset is a Corpus where at a document level the content is text. Each of the document preprocess includes the exclusion of the unicode characters. In the next phase, each of the sentence was split through the process of tokenization. When the tokenization has been completed for all documents , each token Spanish syntax and the category that each token belongs to.

Next, a dictionary feature extraction was applied to each token that stores the word identity, word suffix, word shape

.

and word post tag and the next and previous term sequence.

4. Conditional random fields

Conditional random fields make use of likelihood, to classify the observed variables into a specific category that they belong to. The crf expresses a probability for each word that is generated from the classifier. The probability with the higher value is the selected label for each word.

$$P(y_1 : n|x_1 : n) = \prod_{i=1}^{n} P(y_i|y_{i-1}, x_{x:n}) \tag{1}$$

Updating the weights of the conditional random fields given the raw data and labels, it returns the correct label normalizing the expressed probabilities. crf defines the conditional probability p(y—x), that is the intersection of probability of the x observed variable to belong to a number of classes divided by the probability to belong to the specific class.

$$\frac{P(y \cap y^{n-1})}{y} \tag{2}$$

The gradient descent maximize the conditional likelihood with the process of training.

5. A2c

Actor Critic We can categorize reinforcement learning into

Algorithm 1 Actor Critic (Kristensen & Burelli, 2020)

Randomly initialize critic network $V_{\pi}^{u}(s)$ and actor network
$\pi^{\theta}(s)$ with weights U and θ initialize environment E
for $episode = 1, Mdo$ **do**
  Receive initial observation state $s_0 fromE$
  for $t = 0, Tdo$ **do**
    Sample action $a_t \pi(\alpha|\mu, \sigma) =$ $N(\alpha|\mu, \sigma)$according to current policy
    Execute action $a_t$and observe reward rand next state s_{t+1} from E
    Set TD target $y_t = r + \gamma *$ $VU_{\pi}(s_{t+1})$
    Update critic by minimizing loss $\delta_t =$ $(y_t - V_{\pi}^{U}(s_t))^2$
    Update actor policy by minimizing loss:
    $Loss = -Log(N(a|\mu(s_t, \sigma(s_t)) *$ δ_t
    $Updates_t \leftarrow s_{t+1}$

two distinct methods based on what the model represents-a value based, and policy based on what the model represents, a value based and policy based-. In policy, it takes action that gives the best value.

$$a* = argmax_a Q(s, a) \tag{3}$$

where s is the state and a is the action. In order to add exploration to the training process, it uses epsilon greedy. If the epsilon is 20%, the selection is great with 80% and random with 20%. In order to calculate the policy, we can make use of the probability given the state of a neural network that outputs the softmax.

$$\pi = (a|s) \tag{4}$$

The policy distribution gives the flexibility to select the action that will increase the reward that the agent receives. This approach is also recommended for continuous action that outputs the parameters of any distribution. As a loss function, it makes use of the gradient. First, it makes use of the total rewards that the algorithm knows from samples.

$$J(\theta) = E[\sum_{t=1}^{T} R(s; a); \pi_{\theta}] \tag{5}$$

So, playing an episode, the total rewards are calculated using the policy of π_{θ}. That is the action. Next, it calculates the derivative of J:

$$P(\sum_{t=1}^{T} R(s; a); \theta) = \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\pi_{\theta}(a_t|s_t) \tag{6}$$

$$\bigtriangledown_{\theta} log(\sum_{t=1}^{T} R(s; a); \theta) = \sum_{t=1}^{T} \bigtriangledown_{\theta} log\pi_{\theta}(a_t|s_t) \tag{7}$$

When we use the value function estimated by a second neural network as the baseline, they named it "actor-critic network". The policy network (π) is the actor and the value network (V) is the critic. To sum up, it makes use of a number of steps collected (state,actions), calculates for a number of steps reward and advantage, go in the direction of gradient:

$$\bigtriangledown_{\theta} J(\theta) = \frac{1}{M} \bigtriangledown_{\theta} \, log\pi_{\theta}(a^i|s^i)A(s^i, a^i) \tag{8}$$

6. Deep Deterministic Policy Gradient

The DDPG is an algorithm that of an actor-critic architecture that makes use of a neural network that updates the policy of the network that depends on a parameter

$$f(F((s, a); \theta)) = exp(tF((s, a); \theta)) \sum_{a' \in A} exp(tF((s, a); \theta)) \tag{9}$$

after the extraction of probability distribution θ the weights of the network $\theta = (W, b)$ are updating using gradient ascent rule. The DDPG is a network that is recommended because it introduces an exploration of the probability distribution of the policy:
$a_t \pi^{D}(s_t; \theta_t) + e$, the e is effect of noise that increase the exploration for the agent until the convergence of the network that a small learning learning rate is introduced. The algorithm makes use of the policy and value function(reward,state). The actor updates it's policy and the update of the critic is making with the use of temporal difference. It is an algorithm that does not guarantee convergence.

The implementation of the algorithm starts with the initialization of the Experience replay memory of size 2000,

Algorithm 2 Deep Deterministic Policy Gradient (Guo et al., 2020)

Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor network $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ
initialize target network Q' and $\mu'$with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$
for $episode = 1, M$ **do**
  Initialize a random process$N$ for action exploration
  Receive initial observation state s_1
  for $t = 1, T$ **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + N_t$ according to current policy and exploration rate
    Execute action a_t and observe reward r_t and next state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Set $y_i = r_i + \gamma Q'(s_{t+1}, \mu', (s_{t+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss:$L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$
    Update the target networks:
    $\theta^{Q'} \leftarrow t\theta^Q + (1-t)\theta^{Q'}$
    $\theta^{\mu'} \leftarrow t\theta^\mu + (1-t)\theta^{\mu'}$

that will populate it with each new transition. Two neural networks is implemented one the actor model for the actor model and one neural network for the actor target. For the critic network was implemented two neural networks two Critic models and two for the Critic target.

The training process starts with testing the artificial intelligence algorithm for a number of actions played randomly, next the actor select the policy that the agent will follow. The agent samples a number of states with reward and action to the memory. Given the state the actor will select the action that the agent should follow. The action is expressed as a number of probabilities and a gaussian noise combined as an exploration rate. After the agent performs the actions the two critic networks takes as input the state and action and return two transition probabilities as outputs. The algorithm keeps the minimum transition probability as a representative approximation for the next state. In math terms the final target of the critic is expressed as $r + \gamma * \texttt{minimum(transition probability)}$. The loss of two critic networks are computed as Critic $Loss = \texttt{MSE Loss(transition } probability_{criticone}) + \texttt{MSE Loss(transition } probability_{critictwo})$ The network is getting update using SGD optimizer. Once every two iterations the network is of actor target is getting update with gradient ascent and polyak averaging step

$$\theta' \leftarrow t\theta' + (1-t)\theta' \tag{10}$$

and the critic target is getting updated with same way.

$$\phi' \leftarrow t\phi' + (1-t)\phi' \tag{11}$$

7. Dataset

The dataset consists of 750 texts about medical cases. Each of them was annotated in separate files that depict exactly the word that describes the symptom and the sequence number in the text. The dataset is available on the link:*https://zenodo.org/record/8388807*. The dataset was developed and created in Barcelona as part of the BioCREATIVE 2023.

Corpus	Labels
'O'	0.168
'SYNTOMA'	0.831

Table 1. Total number of Labels

	precision	recall	f1-score	support
SINTOMA	0.74	0.68	0.71	43877
micro avg	0.74	0.68	0.71	43877
macro avg	0.74	0.68	0.71	43877

Table 2. Crf trainset

8. Result

The dataset was used for training the separate at a document level. Each document is represented in numerical form because of the dictionary-based feature extraction method. For the conditional random fields approach, after the training process, the model was evaluated on the same data. The results are displayed on the table. 2 and for each document the result of actor critic are display in figure 1, 2, 3, 4, 5, 6, 7 and 8. The plots are display the number of correct predicted labels from the agent $tp + tn$ and the incorrect classified $fp + fn$. The average number of correct classified entities over all corpus for the trainset is 83%. The Deep deterministic policy gradient approach has a number of rewards close to 85% reward.

9. Conclusion

To sum up the a discriminative model was tested and two reinforcement learning approaches. The crf did not performed as the reinforcement learning approach on the train set and the ddpg had better results in contrast with a2c that requires a lot of training time to be used in unseen data. The higher learning rate gives better reward on the training data.

10. Future Work

The reinforcement learning accomplished almost the same results as the machine learning model. As a future work can be implemented, more complex algorithms can identify entities in the text. An important aspect is the representation of states that can be converted from text to numbers. The main challenge is the encoding scheme that can improve the cumulative reward distribution and an interesting approach is the representation, even in continuous space, that requires more computing power than the current approach and a lot of memory to store the representative matrices. A multi-label task might raise different results as they play an important role in the model precision.

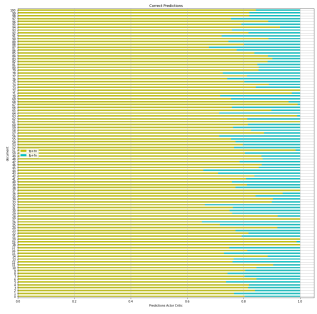

Figure 1. 1-100

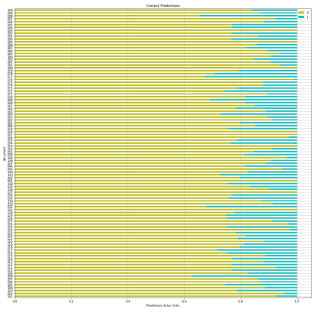

Figure 2. 100-200

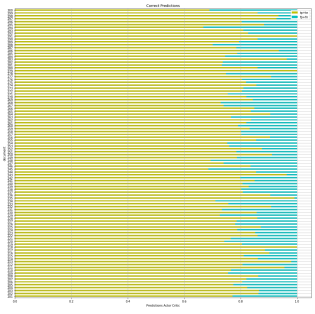

Figure 3. 200-300

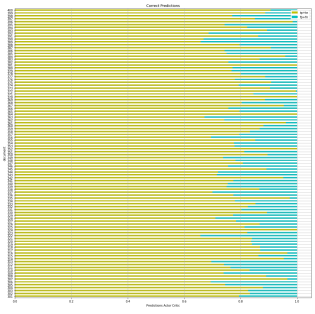

Figure 4. 300-400

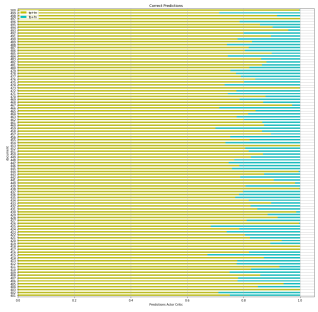

Figure 5. 400-500

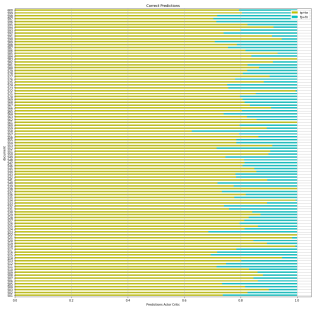

Figure 6. 500-600

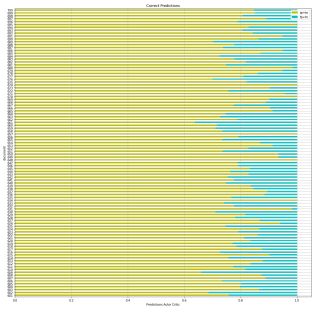

Figure 7. 600-700

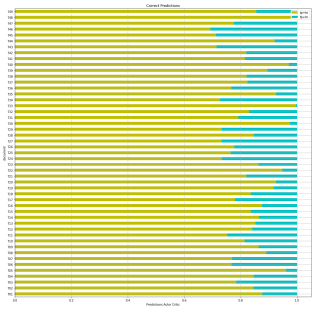

Figure 8. 700-750

References

A.Agrafiots. Medical document anonymization. *Mach. Learn.*, 0-4(0):229–256, may 2019. ISSN 0000. doi: 0000. URL https://www.academia.edu/43457140/Medical_Document_Anonymization.

Guo, B., Zhang, Y.-X., Zheng, W., and Yanhua, D. An autonomous path planning model for unmanned ships based on deep reinforcement learning. *Sensors*, 20:426, 01 2020. doi: 10.3390/s20020426.

Kristensen, J. T. and Burelli, P. Strategies for using proximal policy optimization in mobile puzzle games. In *Proceedings of the 15th International Conference on the Foundations of Digital Games*, FDG '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450388078. doi: 10.1145/3402942.3402944. URL https://doi.org/10.1145/3402942.3402944.

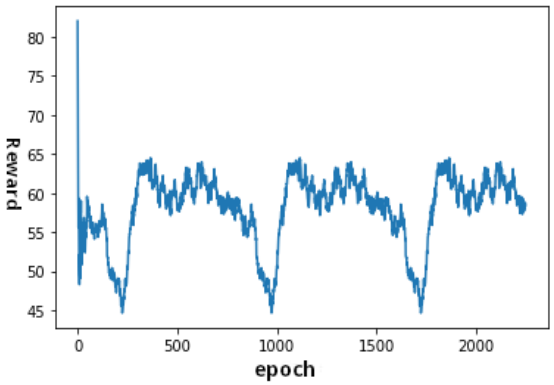

Figure 9. Actor Critic training process $a = 1e - 2500$

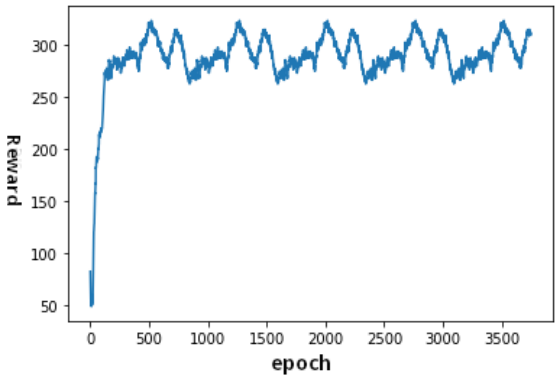

Figure 10. Actor Critic training process $a = 1e - 2$

Figure 11. Actor Critic scores Avg score 81%