A New text classification algorithm based on SARSA in Persian Language

Mehdi, Sadeghzadeh
Computer department, Izeh branch
Islamic Azad University
Izeh, IRAN
e-mail: sadegh_1999@yahoo.com

Hossein, Farzalivand
Young researchers club,
Izeh branch,Islamic Azad University
Izeh, IRAN
e-mail: farzalivand@gmail.com

ABSTRACT

In recent years, with the growing amount of data on information networks such as the Internet, systems require precision and intelligent algorithms for information retrieval. Persian documents are part of this huge volume of information, and information retrieval systems must be designed and created with good efficiency. The deficiency or lack of information is not very important today, but the lack of methods for exploring and exploiting the available information in optimal manner is important. In this paper we present a new approach for searching the document category in Persian documents classified using reinforcement learning algorithms. In this paper, we use the algorithm SARSA, A new approach for exploring text category in classified text that changes its policy. By use of this approach, we have been able to obtain results more relevant and more precise than in other ways of the algorithm and the other algorithms. This study provides encourage results that shows potential of reinforcement learning to solve different problems.

Keywords

Text mining ,reinforcement learning ,SARSA

1. INTRODUCTION

With increasing excessive amount of information, need to develop efficient techniques for intelligent management and retrieval of information is important. Due to the growing need for quick and easy accessing to documents, we should design intelligent retrieval methods to provide contextual information.

Unfortunately, for the Persian language in the field of information retrieval systems due to various problems and lack of coordination, not much has been done. Unfortunately, the lack of national standards in areas such as the Persian alphabetic codes, formats, documentation, storage and retrieval of the existence of problems and lack of coordination between Web services are problems of software developers. Persian on many projects yet to be done is specified requirements and actual requirements of the current work compared with [1].

However, at this time, we believe that everything should be done automatically, even if it was "text understanding".

Other names you will find in this kind of process are: "Text Mining", "Exploring textual data" and 'knowledge discovery in text" [2]. So that analytic review of the literature leads to the formation of the new field of artificial intelligence and information technology has been known as text learning. This field includes all activities about knowledge extraction from text. Analysis of textual data by machine learning techniques, intelligent information retrieval, natural language processing or other related methods are all classified as text learning fields.

However, in today's world, the problem is not the lack of information, but the lack of the knowledge that can be extracted from this information. Millions of web pages, digital libraries, and millions of words on thousands of pages in each company are information resources, but knowledge is summary of the thinking and analyzing about information. Data mining is a very useful way to discover information from structured data the results of whichare stored in tables. Data mining extracts patterns of transactions [3]. However, three main approaches exist in meeting with this huge volume of non-structured information in the world. In formation retrieval [4], information extraction[5] and the discovery of knowledge in this text are three ways to deal with this issue. Document retrieval or information retrieval is the computerized process of producing a list of documents that are relevant to an automatically produced index of the textual content of documents in the system. These documents can then be accessed for use within the same system.

2. RELATED WORK

Several related research projects are investigating the automatic construction of special-purpose web site. The New Zealand digital Library Project [6] has created publicly-available search engine for domains from computer science technical reports to song melodies using manually identified web sources. The CiteSeer project [7] has also developed a search engine for computer science research papers that provides similar functionality for matching references and search. The WebKB project [8] uses machine learning techniques to extract domain-specific information available on the Web into a knowledge base. The WHIRL project [9] is an effort to integrate a variety of topic-specific into a single domain-specific search engine using hand-written extraction patterns and fuzzy matching for information retrieval searching.

Additionally, there are systems that use reinforcement learning for non-spider Web tasks. WebWatcher [10] is a browsing assistant that helps the user find information by recommending which hyperlinks to choose. It, thus, restricts its action space to only hyperlinks from the current page. WebWatcher uses a combination of supervised and reinforcement learning to learn the value of each word on a hyper-link. Our work is not user-centric and strives to find a method for learning an optimal decision policy for locating relevant documents when hyperlink selection is unlimited.

3.REINFORCEMENT LEARNING

Reinforcement learning has become a very active research field in machine learning, and is one of the more recent fields in artificial intelligence. Arthur Samuel [11] was among the first to work on machine learning, with his checkers program. His work didn't make use of the reward signals that are a key component of modern reinforcement learning, but, as Sutton & Barto point out, [12] some of the techniques he developed bear a strong resemblance to contemporary algorithms like temporal difference. Reinforcement learning developed in the early 1990s, generated a lot of interest from the research community. As opposed to the popular approach of supervised learning where an agent learns from examples provided by a knowledgeable external supervisor [13], reinforcement learnin g requires that the agent learn by directly interacting with the system (its environment) and responding to the receipt of rewards or penalties based on the impact each action has on the system.

This paper is organized as follows. The next section introduces a reinforcement learning approach to the search process. In section 5 reinforcement learning algorithms will present. Section 6 consists of the simulation results and discussion. The conclusions are presented in Section 7.

4.REINFORCEMENT LEARNING IN SEARCH PROCESS

In the reinforcement learning framework, a learning agent must be able to perceive information from its environment. The perceived information is used to determine the current state of the environment. The agent then chooses an action to perform based on the perceived state. The action taken may result in a change in the state of the environment. Based on the new state, there is an immediate reinforcement that is used to reward or penalize the selected action. These interactions between the agent and its environment continue until the agent learns a decision-making strategy that maximizes the total reward.

Sutton and Barto [14] defined four key elements for dealing with the reinforcement learning problems: a policy, a reward function, a value function and a model of the environment. A policy defines the agent's behavior in a given state. A reward function specifies the overall goal of the agent that guides the agent towards learning to achieve the goal. A value function specifies the value of a state or a state-action pair indicating how good it (the state or the state-action pair) is in the long run. A model of environment predicts the next state given the current state and a proposed action.

Besides the above four elements, there is a key assumption in the reinforcement learning framework. That is, each decision the agent makes is based on the current state that summarizes everything important about the complete sequence of past states leading to it. Some of the information about the complete sequence may be lost, but all that really important for the future is contained within the current state signal. This is called the Markov property. Therefore, if an environment has the Markov property, then its next state can be predicted from the current state and action. This significant assumption enables the current state to be a good basis for predicting the next state. Under this assumption, the interaction of an agent and its environment can be called a Markov decision process.

For a small reinforcement learning problem, the estimates of value functions can be represented as a table with one entry for each state or for each state-action pair. However, for a large problem with a large number of states or actions, updating information accurately in such a large table may be a problem. Function approximation is currently a popular method to resolve this issue. Function approximation is an approach generalizing experience from a small subset of examples to develop an approximation over a larger subset. Currently, employing neural networks is the most popular approach for function approximation in a large reinforcemen t learning problem[14].

5. REINFORCEMENT LEARNING ALGORITHMS

5.1. SARSA Algorithm

The Sarsa algorithm was first explored by Rummery and Niranjan, who called it modified Q-learning. The name "Sarsa" was introduced by Sutton[15]. The standard procedure of the SARSA algorithm is given as follows:

```
1: Initialize $Q(s, a)$ arbitrarily
2: for all episode do
3: Initialize $s$
4: Choose $a$ from $s$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
5: **for all** step of episode **do**
6: Take action $a$, observe $r$, $s'$
7: Choose $a'$ from $s'$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
8: $Q(s, a) \leftarrow Q(s, a) + \alpha\,[r + \gamma Q(s', a') - Q(s, a)]$
9: $s \leftarrow s'$; $a \leftarrow a'$
10: end for
11: end for
```

SARSA differs from Q-learning in that SARSA is On-Policy, whereas Q-learning is Off-Policy. That means in SARSA, table Q depends on future state-action pair that is chosen by the policy and table Q is updated by these values. So agen ts follow any policy that is used in SARSA algorithm. b Q-learning,Athe highest value available for agen t is updated with table Q, regardless of the policy choice. Each iteration of steps 2-11 represents a learning cycle, also called an ‘‘episode’’. The parameter, α, is the step size parameter and influences the learning rate. The parameter, γ, is called the discount-rate parameter, $0 \le \gamma \le 1$,and impacts the present value of future rewards.

The Q(s,a) values can be initialized arbitrarily. If no actions for any specific states are preferred, then when starting the SARSA procedure all the Q(s,a) values in the policy table can be initialized with the same value. If some prior knowledge about the benefit of certain actions is available, the agent may prefer taking those actions in the beginning by initializing those Q(s,a) values with larger values than the others. Then these actions will initially be selected. This can shorten the learning period. As γ approaches zero, the agent will be more greedy because it takes immediate reward into account more strongly. On the other hand, as γ approaches 1, the agent will be more futuristic. Choosing $\gamma = 1$ is illegal excepet in scenario with finite steps.

5. 2. Exploitation and Exploration

Exploration and exploitation is an important issue when applying the rein forcemen t learn in g algorithm. Exploration means that the agent must try something that has not been done before to get more reward. On the other hands, exploitation is that the agent prefers the actions that were taken before and rewarded. Exploitation may take advantage of guaranteeing a good expected reward in one play, but exploration may provide more opportunities to find the maximum total reward in the long run. One popular approach to deal with this trade-off issue is called an e–greedy method. The e–greedy method involves selecting, with probability (1-e), the action with the best value, otherwise, with small probability e, an action is selected randomly.

5. 3. Policy

In this study, because the algorithm SARSA is based on policy, the agent makes a decision about what work will be done and this makes changes in SARSA agent policy. Popular approach for the exploration and exploitation is the ε-greedy policy. This policy involves choosing theaction with best value and probability (1–ε). Otherwise,an action is selected randomly with a small probability ε. Furthermore, as shown in Table 1, we determine the score an d weight in each areas based on the amount of data that we have on the area, so the probe will be acting according to the scores and upate table Q.

Table 1. Details of classificatio

Categories for Science	Categorie Weight	Scores
Computer	203,666	(9.0 10.0]
Electrical	152,053	(7.0 9.0]
Mechanical	106,135	(5.0 7.0]
Mathematics	13,413	(3.0 5.0]
Chemistry	22,315	(2.0 3.0]
Physics	7,207	(1.0 2.0]

In This Study, All categories are combined into an index which allows rapid searching and retrieval of word. Words have given weights that determine the quality of them in the specified class.

5.4. Naive Bayesian Classifier

The Naive Bayes Classifier has been proven to be a very effective machine learning mechanism and in certain domains it performs better than neural network and decision tree learning. In particular the Naive Bayes Classifier is very well suited to text classification problems.

The Naive Bayes classifier is based on the simplifying assumption that the attribute values are conditionally independent given the target value. That is, the position of the words with relation to each other has no bearing on the classification process.

In general the Naive Bayes Classifier can be applied to learning tasks where each outcome or instance x is described by a conjunction of attribute values and where the target function f(x) can take on any value from some finite set V. In this example the outcome is the classification of a search string, which is dependent on the words (attributes) which make up the search string. The classification is chosen from a limited set of categories V.

In typical text classification problems the trainer is presented with a set of training examples for each target function V. The trainer is then asked choose the most appropriate target for a given test set. The Bayesian approach to classifying the new instance is to assign the most probable target value U given the attribute values (in this case the set of words) $\{a_1, a_2, \ldots, a_n\}$ which describe that instance.

Choosing U involves finding the probabilities of each category being the target value U given the set of words $\{a_1, a_2, \ldots, a_n\}$. We then choose the category with the best probability to become the answer to the classification. Bayes theorem complicates this process a little more by adding a few extra variables into the mix in order to get a more reliable result. In Bayes theorem the formula for 'U' looks something like this:

U = choose best (for each category, calculate $P(a_1, a_2, \ldots, a_n)$)

P = (probability of this category being chosen given $(a_1, a_2, \ldots, a_n)$) * base probability of this category).

The information which gives us the ability to calculate probabilities is drawn from the training data. The frequency of words and categories in the training data provides the weights for the probability calculations. With this assumption, the algorithm looks like this:

U = choose maximum (for each category calculate (base probability of category $* P(a_1, a_2, \ldots, a_n)$)

P = (probability of a_1 indicating this category * probabiulity of a_2 indicating this category ...)

5.5. Reward Function

A reward function defines the goal of the learning agent and determines the value of the immediate action based on the perceived state of the environment. Since the learning agent tries to maximize the total reward, the reward function is then essentially used for guiding the learning agent to achieve its goal. The purpose of this system is finding the nearest thing to a text or document, that will be searched. When a text can be searched by the user, the learner according to the words in the text selects the group with the best probability for each word. Table 2 shows the rewards with the best chance

Table 2. The rewards with the best chance

If (U > MU)
MU = U
Reward = -1
Otherwise
Reward = 1

6. SIMULATION RESULTS

When starting the SARSA algorithm, the values of the state-action pairs, Q(s,a) can be initialized arbitrarily or assigned specific relative values to represent the confidence in favoring each possible alternative. In this study, all the values of the state-action pairs are initialized to zero since all the actions for each state are assumed to be an equally valid choice. This approach starts the system from a neutral state assuming no a priori knowledge of which dispatching rule is best to use in any situation. Therefore, the system would be required to learn from scratch. Other possible alternatives might have been to favor the wrong choice or correct choice initially. The step-size parameter, α which is a small positive fraction, influences the learning rate. The value of this factor can be constant or varied from step to step. In the latter case, the steps become smaller and smaller as learning progresses to assure convergence of Q(s,a) values. With a constant step-size parameter, the Q(s,a) values never completely converge but continue to vary in response to the most recently received rewards. This is more desirable for a non stationary system [14]. The value of the discount-rate parameter, γ, is set between zero and one. As γ approaches zero, the agent is more myopic because it takes immediate reward into account more strongly. On the other hand, as γ approaches 1, the agent will be more farsighted reducing the impact that recent results have on the learned policy. The ε-greedy method is adopted for exploration and exploitation in this study. If ε is set to 0.1, then 10% of the time the strategy will to randomly select one of the three dispatching rules independent of their Q(s,a) values, while the other 90% of the time the dispatching rule with the best Q(s,a) value is selected. Several example systems, such as those illustrated in Sutton and Barto apply the SARSA algorithm with setting of $\alpha = 0.4$ ، $\gamma = 0.9$ and $\varepsilon = 0.1$ This study uses these same common parameter settings for all the experimental runs. Text browser program is implemented with Java Applet on a personal computer and details of data set and experimental documents were shown in tables 3 and 4.

Table 3. Details the data sets used in probe

10 MB	Volume data sets
87,524	Total number of documents
504,789	Number of words
78	Average number of non-repetitive words in each document
126	Average number of words per document

In experiments with a probe to search for 45% of computer science documents are carried out. The results of these experiments are given in Table 5 and Figure 1.

Table 4. Details of the searched documents in probe

100	Number of items
287 KB	Average size of documents
870	Average number of words per document
1045	Maximum number of words in each document
100	Minimum number of words in each document

Table 5. Details of the searched documents in probe

Percent Topic	Field search for robot
45%	Computer
24%	Electrical
18%	Mechanical
2%	Mathematics
3%	Chemistry
8%	Physics

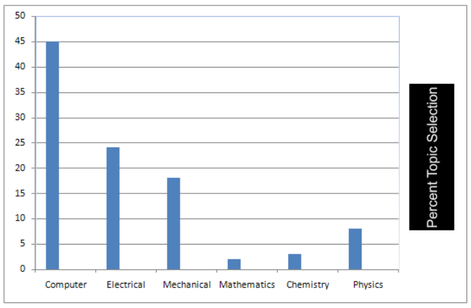

Figure 1. The first tests with the SARSA algorithm and the choice of each area

In two of the experiments conducted to study the issue of documents to search for text classification using SARSA algorithm we propose policy changes. Figure 2 shows one of the experiments conducted with the SARSA algorithm represents a policy change. This experiment was carried out for a document that contains 100 words and SARSA agent near to convergence in episode 15 to almost optimal policy. Figure 3 is one of the SARSA algorithm Experiments showing the change in policy. The test for a text document containing 1045 keywords and SARSA agent is near to convergence in episode 30 to almost optimal policy.

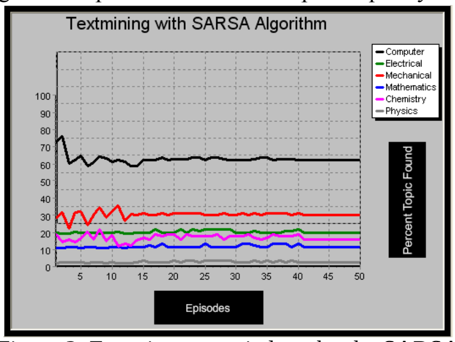

Figure 2. Experiment carried out by the SARSA algorithm with searching 100 words

7. CONCLUSIONS

Reinforcement learning is a very active research area in machine learning. In this study, we use the reinforcement learning algorithms for the search process. Our results confirm that changing the policy SARSA algorithms in search process are able to retrieve relevant information and more precise information. While SARSA algorithm in this environment makes it possible to converge to the optimal policy and agent does not have a very bad experience and never fails. So SARSA algorithmwith changed policy is proposed and implemented in this study and showed better performance than previous approaches using reinforcement learning.

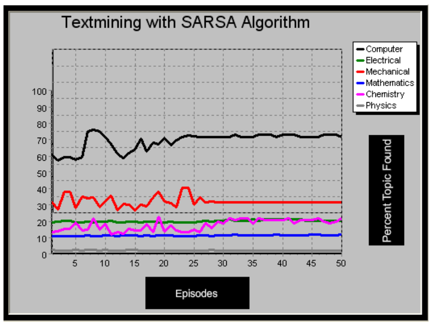

Figure 3. Experiment carried out by the SARSA algorithm with searching 1045 words

REFERENCES

[1] M. A. Hearst. "Untangling text data mining." *Proceedings of the ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics. University of Maryland, June 20-26* , pp. 3-10, 1999.

[2] C. Grover, H. Halpin, E. Klein, Jochen L. Leidner, S. Potter, S. Riedel, S. Scrutchin, and R. Tobin. "A framework for text mining services" *Proceedings of the Third UK e-Science Programme All Hands Meeting (AHM 2004),* pp.878-885 ,2004.

[3] H. Karanikas, dan B. Theodoulidis, "Knowledge discovery in text and text mining software", *Technical report, UMIST - CRIM, Manchester*, 2002.

[4] Y. Kodratoff , "Knowledge Discovery in Texts: A Definition, and Applications," *Foundation of Intelligent Systems, Ras & Skowron (Eds.) LNAI 1609, Springer*, pp. 16-29,1999.

[5] M. Rajman. "Text Mining, knowledge extraction from unstructured textual data" ,*Proc. of EUROSTAT Conference, Francfort (Deutchland),* pp. 473-480,1997.

[6] I. Witten, C. Nevill-Manning, R. Mc-Nab, and S. J. Cunnningham. "A public digital library based on full-text retrieval ", *Collections and experience Communications of the ACM, 41(4)*: pp. 71-75, 1998.

[7] K. Bollacker, S. Lawrence, and C. L. Giles. "An autonomous web agent for automatic retrieval and identification of interesting publications" *Agents '98,* pp.116-123,1998

[8] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell,K. Nigam, S. Slattery."Learning to extract symbolic knowledge from the World Wide Web"*AAAI-98*, 1998.

[9] W. Cohen. "A web-based information system that reasons with structured collections of text," *Agents '98,* pp.400-407,1998.

[10] T. Joachims, D. Freitag, T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web",*Proceedings of IJCAI97*, pp. 770-777,1997.

[11] A.L.Samuel," Some studies in machine learning using the game of checkers." *IBM Journal on Research and Development, 3:* pp. 211-229, 1959.

[12] Sutton, R.S. "Learning to predict by the method of temporal differences" ,*Machine Learning, 3:* pp. 9-44, 1988.

[13] G. Weiss, "Multi agent Systems: A Modern Approach to Distributed Artificial Intelligence." *MIT Press, Cambridge, MA*,1999.

[14] R.S. Sutton, A.G. Barto, "Reinforcement Learning: An Introduction". *The MIT Press, Cambridge, MA*,1999.

[15] G. A. Rummery, and M. Niranjan, "On-line Q-learning using connectionist systems". *Technical Report CUED/F-NFENG/TR 166, Cambridge University Engineering Department*,1994.