# Multi-agent Recommender System: State of the Art

**Conference Paper** · April 2014



**3 authors:**

Afef Selmi
University of Tunis El Manar


Zaki Brahmi
University of Sousse


Mohamed Mohsen Gammoudi
Université de la Manouba

# Multi-agent Recommender System: State of the Art


Afef Selmi
PhD. Student, FST
University of Tunis El-Manar
Tunis EL-Manar, 2092, Tunisia
Email: selmi.afef@yahoo.fr

Zaki Brahmi
Assistant Professor, ISITCOM
University of Sousse
Sousse, 4000, Tunisia
Email: zakibrahmi@gmail.com

Mohamed Mohsen Gammoudi
Full Professor, ISAMM
University of Manouba
Mannouba, 2010, Tunisia
Email: gammoudimomo@gmail.com



***Abstract*—Recommender Systems have become fundamental web applications providing personalized recommendations to users that are suitable to their needs and preferences in order to help them to use interesting resources in a large space of data. These resources may correspond to different types of data and can be distributed over different sites. Then, it becomes difficult to access various resources to retrieve thousands of web pages related to a specific domain, to filter the relevant pages and to integrate these information in order to recommend the best service (web pages, software component, etc.) for users. To solve this problem, cooperatives agents are used. With, extensibility, stability and autonomy of agents characteristics, recommender systems can be more competitive in a dynamic web environment, an open field where data is uncertain. This paper provides an overview of the current works about multi-agent recommender systems.**



***Index Terms*—Recommender Systems, Multi-Agent Systems, Multi-Agent Recommender Systems, Trust.**


## I. Introduction

The first definition of recommender systems has been given by [20]. They describe a recommender system as follows: "In a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases the primary transformation is in the aggregation; in others the systems value lies in its ability to make good matches between the recommenders and those seeking recommendations". This definition has been evolved over time to be generalized by [22] as: "Any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options".

In fact, recommender systems aim to reduce the users search effort by providing personalized recommendations or guiding him to useful resources in a large space of data. These resources can correspond to different types of data such as books [23], scientific articles [7], music [16], travel [8], web pages [2], games [21], etc.

To achieve this goal, a variety of recommendation techniques have been proposed in the literature as [5] [22] [12]. These techniques have been classified into two main categories: content-based recommendation and usage-based recommendation. The first type of recommendation techniques allows analyzing resources content or those descriptions to extract the relevant resources. It consists to identify similar resources to those appreciated by the user. These techniques are precise and applicable when the necessary information is obtained [3]. In addition to this, they are adaptive which means that the quality of recommendation improves over time. However, this recommendations type suffers from the problem of diversity, that is to say only similar resources of those appreciated by users can be recommended [12].

Another problem posed by these techniques is that the recommendation's relevance depends on the user's feedback, in other words there must be a pre-defined number of resources appreciated for the recommendation to be relevant [13]. The second type of recommendation techniques allows providing recommendations to users based on the user's behavior analysis and/or his feedback [1]. These techniques are applicable to any type of data. However, to provide relevant recommendations; the system must be manipulated several times by the user; this problem is called "Cost Start Problem" [12]. Also, in this type of recommendations, the percentage of recommendations appreciated by users is low compared with the volume of data [13].

To overcome these problems, current works are oriented to an intelligent research which aims to satisfy the best needs and preferences of users based on agents techniques [8] [9] [10] [11] [2] [21]. With extensibility, stability and autonomy of agent's characteristics, recommender systems can be more competitive in a dynamic web environment because they automatically adapt to the changes of their environments. In these systems, agents can operate in a cooperative way; they are able to deliver to user only relevant and personalized recommendations [11]. In this paper, we present a state of the art of multi-agent recommender systems. Following a comparative study, we have identified some problems related to these systems which represent the main subject of our work.

The rest of the paper is organized as follows: the next section gives an overview of the various recommendations techniques. In section 3, we study, in a fist part, the most recent multi-agents recommender systems. Then, a comparative study of various studied works is presented. We expose our problem in the part discussion of section 4. We end this paper with a conclusion and our future works.

## II. Recommendation Techniques

Recommendation techniques have been classified in different ways [5] [22] [12]. A first classification consists in grouping these techniques into two main categories, namely content-based recommendation and usage-based recommendation [5] [12]. A second classification consists of three other techniques such as demographic data-based recommendation, utility-based recommendation and knowledge-based recommendation [22]. These techniques are considered as special forms of content-based recommendations [13].

### A. Content-based recommendation

The principle of this technique is based on resources content analysis or those descriptions to extract the relevant resources. It consists to identify similar resources to those appreciated by the user according to their content. This type of recommendation needs to identify resource profiles and user profiles. Resource profiles consists on a set of attributes describing the resources in the same way that an index in information retrieval field. Stemming technique [18] is used to determine the resource profiles. First, it consists to transform words related to the same concept in a same term that represents all. Second, it consists to attribute a weight to each of these terms according to their importance in the resource using the tf.idf formula (term-frequency times inverse document frequency) [14]. User profiles consist of a set of information that can be manually input by the user or automatically extracted from resource contents accessed by the user. The extraction of user profiles is done by two ways: the first way is to extract information from a completed form by the user providing its interests. The second way is an automatic extraction of resource contents accessed by user.

There are three approaches which are considered as particular forms of content-based recommendation. We detail in the following section each of these approaches.

**Demographic data-based recommendation:** consists in distributing users into different classes according to their demographic data such as sex, age, occupation, etc. [22] [3]. The main idea with this technique is that if two users have been evolved in a similar environment then they are more likely to have common preferences. As advantages, this recommendation technique is applicable when the necessary information is obtained. In addition, it provides relatively relevant recommendations when the user starts using the system. However, for raisons of respect for private life, it is not always possible to get all users information.

**Utility-based recommendation:** also called preferences-based recommendation, it determines recommendations according user's utility function [19]. It is to request users to complete forms in order to extract information about a product they are buying for example. The problem proposed by this technique is the definition of the utility function.

**Knowledge-based recommendation:** consists in regrouping user's information in order to recommend him resources [4]. It allows also exploiting links between resources: for example use knowledge of cuisines to predict the similarity between restaurants. This technique needs three types of knowledge: catalog knowledge (knowledge about objects and their characteristics to recommend), functional knowledge (the system should be able to match user's needs with objects that could meet its needs) and user knowledge (the system must have some knowledge about users to be able to provide relevant recommendations).

### B. Usage-based recommendation

Compared to content-based recommendation techniques, usage-based recommendation techniques (usages are considered as explicit attribution of appreciations resources by users) don't take into account resources content which avoids the extraction of resource profiles and user profiles. In addition, these techniques are independent recommendation of data nature. Its principle is based on the behavior analysis and/or user appreciations. There are two recommendation strategies: the first strategy exploits user appreciations to recommend resources that have been appreciated by other users with similar preferences. The second strategy detects user behavior patterns to predict resources that are likely to interest a particular user. Two main techniques for this type of recommendation are proposed: collaborative recommendation and patterns[1] detection. The principle of each technique is detailed in the following.

**Collaborative recommendation:** the most used technique [1], is based on the exploitation of user appreciations of resources. These appreciations are represented by notes[2] that are explicitly or implicitly attributed by users. This recommendation technique aims to predict notes that users will attribute to resources to recommend then resources with the best predicted note. These notes are represented by a matrix whose rows represent users, columns represent resources and cells represent the users attributed note to the resource. The collaborative recommendation is classified into two approaches namely memory approach and model approach. The first approach calculates each rating prediction from the matrix using weighted sum of these notes. It is simple and efficient; it allows a dynamic adaptation of new notes entered in matrix. However, it suffers from a problem of complexity and scalability, also its use is possible only in a small data space [13]. The second approach solves the problem of complexity posed by the memory approach using users models and resource's models. These two models are constructed off-line; they will be later used to calculate recommendations. For this approach, the calculation of rating prediction is done in two ways: the first builds a probabilistic model to determinate the most score, the second groups users or resources in homogeneous sub-groups then applies the memory approach into the sub-groups obtained.

**Pattern detection:** this recommendation technique calculates recommendation and presents to the user resources which he likes to consult later. Association Rules is an approach of pattern detection. Indeed, the association rules extracted, using Apriori algorithm, represent a model that can be used later to predict resources [15]. To predict resources, given a

[1] A pattern is a sub-set, a sub-sequence or sub-structure which is repeated in a data space [15].

[2] A note is a numeric value representing the user appreciation to a resource.

user history, just compare all antecedents of user history with antecedents of rules extracted. If an antecedent corresponds, then, the result can be recommended. If more antecedents correspond, then, it is possible to sort obtained rules according to their trust and/or their support. The association rules are extracted from all historical consultation of all users.

### C. Comparing recommendation techniques

We present in this section a comparative study of recommendation techniques such as content-based recommendation and usage-based recommendation. Table 1 shows the advantages and disadvantages of each of these recommendation techniques.

TABLE I
COMPARING RECOMMENDATION TECHNIQUES

| **Techniques** | **Content-based recommendation** | **Usage- based recommendation** |
|---|---|---|
| Advantages | - Precise techniques,<br>- The domain knowledge isn't obligatory,<br>- Demographic data-based recommendations applicable when the necessary information is obtained,<br>- Adaptive techniques: recommendation quality improves over time. | - Applicable to any data type,<br>- Doesn't consider the resource content,<br>- Techniques independent of data nature. |
| Disadvantages | - Doesn't consider users appreciations, they consider only resources content,<br>- Are applicable only on textual data,<br>- For raisons of respect for private life, it is not always possible to get all users information,<br>- The extraction of resource profiles depend on data nature,<br>- For the extraction of user profiles, it's possible that the user can't complete the form,<br>- Problem of diversity: content-based recommendation techniques can only recommend that similar resources to those appreciated by users. | - Techniques dependent on user feedback,<br>-Collaborative recommendation suffers from the "cold start problem", means that the system can provide relevant recommendation when user accords appreciations for a number of resources,<br>- Pattern detection technique is only applicable in a reduced space of data. |

Recommender systems have been defined as tools that provide recommendations of users to meet their needs and preferences. To achieve this goal, recommender systems have used information available on the web which is a rich data source in which the user can find information that meets his needs [11]. However, the data volume causes some problems such as the overload of data, heterogeneity of information, the dynamic change of information, the maintenance problem, etc. Indeed, in [11], authors assert that the problem of recommending a better service to users becomes a major problem. Moreover, as shown in the famous example in the tourism field, to recommend traveling to user, the travel agent must know the set of services (travel, hotel, programs, etc.) to compose travel according to user preferences. This task isnt always evident because services are distributed on different resources. Thus, allocation of information resources, access, filtering and integrating information to compose a travel became a problem that attracts many researchers [10].

To reply this problem, recently, many multi-agent recommender systems have been proposed in the literature [11] [8] [2] [21]. We are interested in the following to multi-agent recommender systems.

## III. Multi-agent Recommender System

Current works are oriented to an intelligent research which aims to satisfy the best needs and preferences of users using multi-agent approach to develop interactive systems. In these systems, each agent independently performs a specific task and different agents operate simultaneously to accomplish overall tasks. In a multi-agent system, agents are extensible (agents are independent, they can be created or modified, other agents continue to provide services), stable (when an agent is out of services, other agents share tasks and ensure the continuity of services) and independent (they are running without user intervention). With extensibility, stability and autonomy of agent's characteristics, systems can be more competitive in a dynamic web environment because they automatically adapt to environments change.

We are interested, in our work, to multi-agent recommender systems. They have been used in several application areas such as games domain [21], information retrieval domain [2], tourism domain [8], legal domain [17], etc.

### A. Tourism domain

In [11] [10] [9] and [8], authors presented the evolution of their recommender system. Indeed, in [11], the author proposed a multi-agent recommender system to recommend a travel to a user according to his needs. The architecture of this system includes two types of agent that share a common goal (the recommendation) and assigns a separate individual goals (research services). These agents communicate together by exchanging messages. According to authors, the proposed system provides good recommendations to users according to their needs and preferences. However, these agents are not specialists for all package services. Also, their knowledge base is centralized.

Reasons for [10] have proposed a new multi-agent recommender approach based of distributed knowledge: knowledge about customers, about services, about application domains. Indeed, resources can be distributed to several agents who work as experts, cooperate with each other by exchanging information stored in their knowledge base. In this approach, author has proposed a Trust Maintenance System (TMS) to keep the integrity of shared data and the integrity of agent knowledge bases. Each agent of this system computes a trust degree each time it chooses a task to run. After running, the agent checks if the value of this degree has increased or not. According to this value, it decides to keep the task in its knowledge base or not. This trust degree is determined by the following formula:

$$confind' = \frac{(confind + F(typet, a))}{\sum_{typeti \in KB} F(typeti, a)} \quad (1)$$

Where ***F(typet,a)*** represents the frequency of the task ***typet*** performed by the agent. The frequency is calculated by the number of times that the agent performs this task which is equal to the sum of all tasks performed by the agent.

With this trust degree, the author [10] tried to improve the recommendation process defining a new approach based on distributed knowledge among different agents of the community. These agents are able to maintain and evaluate the integrity of information exchanged between them with the new TMS component (Trust Maintenance System). However, the communication between agents in this system can generate some problems, namely the contradiction of knowledge exchanged and the heaviness of response because the number of messages exchanged between agents.

An extension of previous works in [9] consists on proposing a new trust model for multi-agent recommender system. The objective of this model is to avoid unnecessary communication: communication with agents that haven't the information requested in order to improve the information process exchanged between agents. In fact, in the MATRES system (Multi-Agent Travel Recommender System), the recommendation cycle starts when user defines a query by specifying all its preferences. The user query will be divided into several tasks (L = $t_1$, $t_2$... $t_n$ is the set of tasks). These tasks are processed by different agents of the community (C = $a_1$, $a_2$... $a_n$ represents the community of agents). Each agent calculates the trust degree (which measures the efficiency of the previous cooperation between agents) of all other agents in the community C in order to choose a trust agent to exchange with him the necessary information for the running of its tasks. The trust degree of an agent $a_i$ for an agent $a_j$ with respect to tasks of type $t_k$ is given by the following formula:

$$T^{t_k}_{a_i,a_j} = \frac{\sum_{\theta=1}^{\eta} \vartheta_{\theta}^{t_k} \times e^{-\tau\theta}}{\sum_{\theta=1}^{\eta} e^{-\tau\theta}} \quad (2)$$

- $\vartheta_{\theta}^{t_k}$: represents the average of the positive evaluations for the task $t_k$ solved at time $\theta$, this average varies between 0 and 1,
- $\tau$: represents a constant that weights the value $\vartheta_{\theta}^{t_k}$ when updating the trust degree. It is defined according to the number of days ($\theta$) in the equation $e^{-\tau\theta}$,
- $\eta$:represents the number of days passed since the last evaluation,
- This formula returns a value in [0,1] where 0 is the minimum trust degree and 1 is the maximum trust degree.

In [9], authors have proposed a trust model for multi-agent recommender system. With this trust degree, each agent will be able to choose a trust agent to exchange with him information necessary to run its task allowing to prevent unnecessary communications between agents and allowing agents to be experts in solving certain types of tasks during the recommendation cycle.

The same authors have defined the MATRES system (Multi-Agent Travel Recommender System) in [8]. This system aims to provide a best recommendation based on assumptions determined by agents where there is a delay for receiving the missing information. To ensure the correct behavior of system, the user evaluates recommendation to be later stored in the knowledge base of agents. Indeed, the multi-agent recommender system MATRES is a new application that recommend various travel services for user according to their preferences in each service. In addition, this system uses a trust mechanism to improve information exchange process between agents. However, in MATRES, the number of attributes, defined by an expert agent, involved in the recommendation process can cause a significant number of assumptions. Moreover, the trust degree used in this approach hasn't given good results [8]. As future works, authors have proposed to extend this mechanism by exchanging assumptions between agents which improves system performance because agents are able to use assumptions of other agents instead to do it themselves.

### *B. Information retrieval domain*

Today, several organizations have resorted to use web in order to facilitate information delivery. But, with the large number of web pages, user will find a difficulty to consult all pages to extract information that meets their needs. They also lose a lot of time doing research. Adapting web problem with user needs isn't a new problem [6]. Thus, authors have proposed using multi-agent approach, since it is flexible and able to be adapted dynamically with web applications.

In their works, authors [2] have treated the same adapting web problem. They have proposed a multi-agent recommender system of web pages to users. The principle of this approach is based on the combination of two algorithms which works with binary data: AR algorithm (Associative Rules) and CF algorithm (Collaborative Filtering). Indeed, in this system, two recommendation agents were created: the first uses the AR algorithm to generate a single condition of association rules; the second uses the CF algorithm. By combining these two algorithms, recommendations provided by this system are designed to be faster in order to keep user's interests and minimize response time. In addition, the proposed multi-agent recommender system takes into account that each agent must quickly respond to any request from another agent, and must prepare in advance for the next request and for tasks that require a lot of time.

### *C. Games domain*

Since the number of games available to user increases from one day to another, it is difficult to find games that are best suited with user's competences. It is necessary, therefore, to adapt the difficulty of games with user competences and their abilities. The aim of Pavlov and al. project [21] at Croatia University is to develop an intelligent and adaptive tool with user competences. In fact, authors have proposed a multi-agent recommender system of games for mobile phones. The proposed system is to recommend games which adapt better with user competences and their abilities in order to maintain longer and increase the time playing some games. The system architecture comports four types of agent: an UA agent (User Agent) which is responsible for supervising and analyzing the user interaction with games, a SPA agent (Service Provider Agent)represents the service provider and is responsible for recommending games according to the category and the difficulty specified in the query of UA agent, a DA agent (Database Agent) which is responsible for updating the SPA agent's database, a CPA agents which are responsible for the production and the publication of games.

## IV. Discussion

Table 2 shows a comparative study of multi-agent recommender system against the following criteria:

- Application area: it is to specify in which area the proposed system has been applied,
- Knowledge organization: it is to check if knowledge is centrally organized or distributed,
- Trust modeling: it is to check if the trust has been modeled or not. If so, how authors have evaluated this trust.
- Evaluation of recommendation: it is to check for each system, if the user has evaluated the recommendation provided by the system or not.

TABLE II
COMPARING MULTI-AGENT RECOMMENDER SYSTEM

| | Domain | Knowledge organization | Trust modeling | | Evaluation of recommendation |
|---|---|---|---|---|---|
| | | | Modeling | Evaluation | |
| [11] | Tourism | centralised | - | - | - |
| [10] | Tourism | distributed | Compute the trust degree for each task | confind'= $\frac{(confind+F(typet,a))}{\sum_{typeti\in KB} F(typeti,a)}$ | |
| [9] | Tourism | distributed | Compute the trust degree between agents | $T^{t_k}_{a_i,a_j} = \frac{\sum_{\theta=1}^{\eta} \vartheta^{t_k}_{\theta} \times e^{-\tau\theta}}{\sum_{\theta=1}^{\eta} e^{-\tau\theta}}$ | - |
| [8] | Tourism | distributed | - | - | yes |
| [2] | Information retrieval | distributed | - | - | - |
| [21] | Games | distributed | - | - | - |

In our comparative study of different recommender approaches, we can conclude that:

- Multi-agent recommender systems are still dedicated to their application domains.
- In [9], authors have proposed a trust model for multi-agent recommender system. However, they didn't consider a network of trust, that is to say, when an agent $a_i$ trusted an agent $a_j$ but it doesn't have the information then it will be able to able to request another agent in network.
- In the system proposed by [8], agents can define assumptions in the case when there is a delay in receiving missing information, which is necessary for recommendation. But, they didn't address the reusability of assumptions that is to say the exchange of assumptions among agents to solve the delay problem.
- All works presented haven't studied agent's behavior that may change according to the context and resources that may also change over time. In addition, they haven't studied the dynamic change of user behavior and the evolution of user preferences over time.
- These approaches are mainly used for content web pages research not to search for software components as *Web Services* and *Cloud Computing* which represent current topics. With increasing presence and adoption of web services and cloud computing on the World Wide Web, Trust is becoming important; as crucial for users as for services provider. Then, it becomes necessary to think about modeling and evaluation of this trust especially in an open environment (Web Services, Cloud Computing, ...), in a dynamic environment where data or services provider are uncertain.

## V. Conclusion

We have presented, first, in this paper a state of the art of recommendation techniques. Second, we have compared these techniques to identify advantages and disadvantages of each one. We have focused, thereafter, on multi-agent recommender systems. A comparative study was developed at the end of this paper to address problems related to the presented multi-agent recommender system.

To solve some limitations of these systems, we have thought to develop a generic recommendation approach based on the extension of multi-agent approach in large-scale environment by working on the following points:

- Discovery: this is to identify the different agents of system having relevant information, to identify the better data source for the query and to identify services as well as information provided by agents.
- Evolution: the agent behavior may change according to the context; resources can also change over time. It seems also useful to think of the evolution of user preferences over time. At this level, we think it is necessary to define a new approach able to detect the change of use behavior and to adapt recommendations to the new needs of users.
- Trust: because we are in a highly dynamic environment where data are uncertain, the trust is a key point. Then, it will be possible to propose a trust model to be able to decide in a large-scale environment. With this feature, the agent is more likely to find relevant information.
- Reusability: we think that assumptions mechanism must be extended by exchanging assumptions between agents, which improves system performance because agents

would be able to use assumptions of other agents instead of defining them by themselves to avoid any loss of time.

## REFERENCES


[1] A. Das, M. Datar, A. Garg and S. Rajaram: 'Google News Personalization: Scalable Online Collaborative Filtering', WWW'07, Proceedings of the 16th International Conference on World Wide Web, pages 271280, 2007.

[2] A. Jorge Morais, Eugénio Oliveira and Alipio Mario Jorge: 'A Multi-Agent Recommender System', In International Symposium on Distributed Computing and Artificial Intelligence, Salamanca, 2012.

[3] A. Nguyen, N. Denos and C. Berrut: 'Exploitation des données disponibles à froid pour améliorer le démarrage à froid dans les systèmes de filtrage d'information', Informatique des Organisations et Systèmes d'Information et de Décision, pages 8195, 2006.

[4] B. Towle and C. Quinn: 'Knowledge Based Recommender Systems Using Explicit User Models', Pages 7477, 2000.

[5] C. Shahabi, F. Banaei-Kashani, Y. Chen and D. Mcleod: 'Yoda: An Accurate and Scalable Web-Based Recommendation System', In Sixth International Conference on Cooperative Information Systems, pages 418432, 2001.

[6] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich: 'Recommender systems: an introduction', Cambridge University Press, 2011.

[7] D. Pavlov, E. Manavoglu, D. Pennock and C. Giles: 'Collaborative Filtering with Maximum intelligent Systems', 19(6):4048, ISSN 1541-1672, 2004.

[8] F. Lorenzi, Ana Lucia, Cetetich Bazzan, Francesco Ricci and Mara Abel: 'Improving recommendations through an assumption-based multi agent approach: an application in the tourism domain'. Expert Systems with Applications, 2011.

[9] F. Lorenzi, G. Baldo, R. Costa, M. Abel, A. Bazzan and F. Ricci: 'A Trust Model for Multi-Agent Recommendations', Journal of Emerging Technologies in Web Intelligence, 2010.

[10] F. Lorenzi: 'A multi agent knowledge-based recommender approach with truth maintenance', RecSys07: Proceedings of the ACM conference on Recommender systems, 2007.

[11] F. Lorenzi, A. Bazzan and M. Abel: 'An Architecture for a Multi-Agent Recommender System in travel Recommendation Scenarios', Proceedings of 3rd Workshop on Model-Based Systems, ECAI 2006.

[12] G. Adomavicius, R. Sankaranarayanan, S. Sen and A. Tuzhilin: 'Incorporating contextual information in recommender systems using a multidimensional approach', ACM Transactions on Information Systems, 23(1):103145, 2005.

[13] Geoffray Bonnin : 'Vers des systémes de recommandation robustes pour la navigation Web: inspiration de la modélisation statistique du langage', Thése, Université Nancy 2, 2010.

[14] G. Salton and C. Buckley: 'Term Weighting Approaches in Automatic Text Retrieval', Rapport technique, Ithaca, NY, USA, 1987.

[15] J. Han and M. Kamber: 'Data Mining: Concepts and Techniques (Second Edition)', Morgan Kaufmann, second edition, 2006.

[16] J. Su, H. Yeh, P. Yy and V. Tseng: 'Music Recommendation Using Content and Context Information Mining'. IEEE Intelligent Systems, 25(1):1626, 2010.

[17] L. Drumond, R. Girardi, and A. Leite: 'Architectural design of a multi-agent recommender system for the legal domain', In ICAIL: Proceedings of the 11th international conference on Artificial intelligence and law. New York, NY, USA: ACM, 2007.

[18] M. Porter: 'An Algorithm for Suffix Stripping', Pages 313316, 1997.

[19] M. Stolze and R. Walid: 'Towards Scalable Scoring for Preference-based Item Recommendation', IEEE Data Engineering Bulletin, 24:4249, 2001.

[20] P. Resnick and H. R. Varian: 'Recommender systems', Commun, ACM 40, 3, 56-58, 1997.

[21] P. Skocir, L. Marusic, M. Marusic and A. Petric: 'The MARS A Multi-Agent Recommendation System for Games on Mobile Phones', Agent and Multi-Agent Systems, Technologies and Applications Lecture Notes in Computer Science Volume 7327, pp 104-113, 2012.

[22] R. Burke: 'Hybrid Recommender Systems: Survey and Experiments'. User Modeling and User-Adapted Interaction. 12, 4, 331-370, 2002.

[23] R. Mooney and L. Roy: 'Content-based Book Recommending using Learning for Text Categorization', In Proceedings of the fifth ACM conference on Digital libraries, pages 195204, 2000.