An Agent based Platform for Resources Recommendation in Internet of Things

Agostino Forestiero[a] and Giuseppe Papuzzo[b]
Institute for High Performance Computing and Networking, National Research Council of Italy, Via P. Bucci 8-9C, Rende (CS), Italy

Keywords: Multiagent Systems, Internet of Things, Word Embedding, Recommendation Systems.

Abstract: Internet of Things (IoT) paradigm aims to bridge the gap between physical and cyber world allowing a deeper understanding of users in terms of preferences and behaviors. User devices and services interact and maintain relations which need of effective and efficient selection/suggestion approaches to better meet users' interests. Recommendation systems provide a set of significant and useful suggestions for users and systems with given characteristics. This paper introduces the design of an agent based platform for building a distributed recommendation system in IoT environment. Internet of Things objects (devices, sensors, services, etc.) are represented with vectors obtained through the *Doc2Vec* model, a neural word embedding approach able to capture the semantic context representing documents in dense vectors. The vectors are managed by cyber agents that, performing simple and local operations, organize themselves exploiting the vector values. The outcome is the emergence of an organized overlay-network of cyber agents that allows to obtain an efficient recommender system of IoT services. Preliminaries results confirm the validity of the approach.

1 INTRODUCTION

Several objects interconnected with each other achieving a common purpose represent the Internet of Things (IoT) paradigm (Atzori et al., 2010). The number of objects and the amount of data generated by IoT infrastructures are growing hugely and this makes the traditional manage mechanisms inadequate. Intelligent and automated approaches are needed to support decision makers due to the dynamic nature of smart objects, devices and services, involved in the IoT systems. Systems able to perform "things recommendation" is a crucial step to take full advantage of the IoT. Recommendation systems are an important research topic and several works have been proposed both in the industry and academia. These systems allow to individuate a list of useful items for the users in a given context. The usefulness of an item or product or service is generally represented by a "rating", which indicates *how much a given user likes a particular item*. The items with a high rating value are presented as recommendations for the user.

Recommendation systems can be categorized as (Balabanović and Shoham, 1997): (i) *Collaborative Filtering* (CF), an item is recommended to the user according to the past ratings of all users. The approach evaluates the utility of the item i for the user u by estimating the usefulness assigned to item i by the users v who are "similar" to user u; (ii) *Content-based recommending* an item is recommended if it is similar to items that the user has chosen in the past. Information retrieval (IR) techniques address this problem, where the content associated can be handled as a query, and the unrated documents marked with a similarity value to this query. Otherwise, the documents can be converted into word vectors, and then averaged to obtain a prototype vector of each category for a user, as reported in (Lang, 1995); and (iii) *Hybrid approaches* in which collaborative and content-based approaches are combined. The similarity between two users can be computed using various approaches, but the most popular are correlation (Resnick et al., 1994) and cosine similarity (Breese et al., 1998). Collaborative and content-based approaches use the same cosine measure from information retrieval. But, in content-based recommendation systems measures the similarity between vectors of weights, whereas, in collaborative systems measures the similarity between vectors of the actual ratings specified of the users. Other approaches to the recommendation consist in handling

[a] https://orcid.org/0000-0002-3025-7689
[b] https://orcid.org/0000-0003-3961-3966

Forestiero, A. and Papuzzo, G.
An Agent based Platform for Resources Recommendation in Internet of Things.
DOI: 10.5220/0009573207750779
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 775-779
ISBN: 978-989-758-423-7

of the problem as a classification task. Each pattern represents the content of an item, and a user's past ratings are used as labels for these patterns. For example, text from fields such as title, author, synopses, reviews, and subject terms are used by (Mooney and Roy, 2000) to recommend books.

Several classification algorithms have been also used to content-based recommend: decision trees, k-nearest neighbor, and neural networks (Pazzani and Billsus, 1997). The heterogeneity of possible scenarios, arising from the massive deployment of an enormous amount of smart objects, imposes the use of sophisticated and innovative models and algorithms. We propose an advanced version, enriched with semantic properties, of the agent-based algorithm for building a *things* recommendation system introduced in (Forestiero, 2017), in which bio-inspired agents work together in order to obtain a common purpose that is the organization of distributed resources. Agents are: *extensible*, they can be created or modified; *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 characteristics as extensibility, stability and autonomy, they can be more adequate in a dynamic system because they automatically adapt to environments change (Selmi et al., 2014).

The algorithm proposed is able to organize *things* of an IoT environment in order to improve recommendation operations. In particular, each smart object is associated with a "cyber agent", which represents it in a *cyber layer*. The cyber layer is a virtual layer in which the cyber agents can"collaborate" among them, in a peer to peer fashion (Forestiero et al., 2008a), in order to obtain a common goal and improving the performances of the system (Forestiero et al., 2008b)(Forestiero et al., 2005). *Vectors* of real numbers, are exploited to describe IoT objects. The vector can have different meanings, for example: the presence or absence of a given characteristic or it can be the result of a hash function locality preserving so that similar vectors are assigned to things with similar characteristics. In peer to peer systems, indeed, metadata representing the content are often indexed through bit vectors, or *keys*, which can have different meanings. One is that each bit represents the presence or absence of a given *topic* (Crespo and Garcia-Molina, 2002) (Platzer and Dustdar, 2005): this method is particularly adapt for contents like documents, because it is possible to identify the different topics existing in the documents. Alternatively, a metadata can be mapped through a hash function into a binary vector. The hash function have to be designed locality preserving (Cai et al., 2004) (Oppenheimer et al., 2005), thus, neighbor vectors are assigned to contents with neighbor/similar characteristics. Similarity measure can be the cosine of the angle or the Euclidean distance between the bit vectors.

In our approach, the *Doc2Vec* model (Le and Mikolov, 2014), able to represent documents in dense vectors, also capturing the semantic, is exploited to map smart objects. The cyber agents organize themselves based on the similarity of its vector with the wished IoT device/service. The outcome of the algorithm is a logically sorted list of cyber agents based on the similarity with the target IoT device/service, where the distance from the target IoT device/service increase with the distance from the initial position of the list. Thanks to this organized list, the suggestion operations become faster, because we can find similar, probably useful and recommendable vectors (smart objects) in the first positions of the list.

2 SMART RECOMMENDATION ARCHITECTURE

The aim is to design a platform able to provide useful resource suggestions in Internet of Things. To achieve this objective, a semantic multiagent algorithm was designed and implemented. Physical devices, sensors, services, etc. are represented by *vectors* obtained through the *Doc2Vec* neural model (Le and Mikolov, 2014) applied to the metadata (text) describing them. *Doc2Vec* is an unsupervised algorithm to generate vectors starting from sentences/documents based on *Word2Vec*, a word embedding approach which can generate vectors starting from words. *Word2Vec* (Le and Mikolov, 2014) is a two layer artificial neural network used to process text to learn relationships between words within a text corpus. Word2Vec takes as its input a large corpus of text and produces a high-dimensional space (typically of several hundred dimensions), with each unique word in the corpus being assigned a corresponding vector in the space. This "word embedding" approach is able to capture multiple different degrees of similarity between words. To create the model of relationships between the words, a particular grouping of text or documents is fed to the *Word2Vec* process, which is called the training corpus. *Word2Vec* builds a vocabulary exploiting a corpus and, by training a neural network with three levels, learns the word representations. *Word2Vec* proposes two kind of models: (i) Continuous Bag of Words (CBOW) that learns the representations by predicting the target word based on its context words; and (ii) Skip-gram, that learns representations by predicting each of the context words

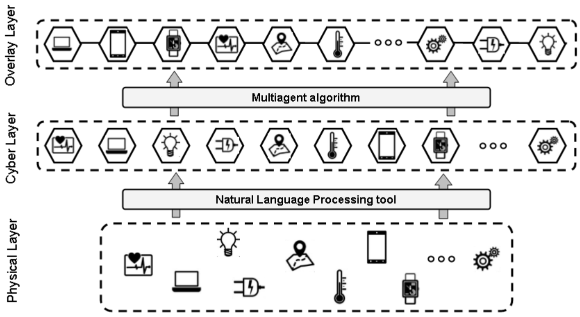

Figure 1: Logical layers of the infrastructure. The Overlay Layer depicts the outcome of the algorithm.

based on the target word. So, one has to choose one of the architectures and set values for hyper parameters like embedding size, context size, minimum frequency for a word to be included in the word vocabulary to generate the word embeddings from a large corpus of unlabeled data. The distance/similarity between two IoT devices/services can be computed through the cosine distance/similarity between the vectors representing the description. Given two IoT devices/services, the cosine measure utilized to compute the similarity between them is reported in formula (1). The cosine-based method uses two vectors in *n-dimensional* space to represent the users u and v, and n will be $|I|$. The cosine of the angle between two vectors, as reported in formula (1), can be computed to measure the similarity between them, where $\vec{u} \cdot \vec{v}$ is the *dot-product* between the vectors $\vec{u}$ and $\vec{v}$.

$$Similarity(\vec{u}, \vec{v}) = cos(\vec{u}, \vec{v}) =$$

$$\frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \times |\vec{v}|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}} \quad (1)$$

Vectors obtained with *Doc2Vec* library and describing IoT devices/services, are assigned to cyber agents that, in fully distributed and self-organizing modality, exploiting only local information, organize themselves in order to improve recommendation operations. The outcome of the algorithm is a sorted overlay network of cyber agents, organized on the basis of the similarity with the wished IoT device/service.

Figure 1 reports a logical architecture of the algorithm. The infrastructure is organized in three main layers: (i) a physical layer, composed of a collection of IoT devices/services connected among them through local area networks and able to collect data and information coming from the physical world; (ii) a cyber layer, consisting of a group of cyber agents representing the IoT devices/services. In this layer, the cyber agents work and collaborate together following the steps of the algorithm in order to produce an overlay network able to provide useful and relevant information for recommendation operations; and (iii) an overlay layer, in which an overlay network of cyber agents emerges as outcome of the algorithm. The topology of the emerging overlay depends on the strategy of the algorithm performed by the cyber agents. In this figure, a similarity ordered list of cyber agents emerges. This list allows to select a given number of cyber agents, representing IoT devices/services, that can be suggested to the user and probably useful and relevant because they are very similar to the wished IoT device/service.

2.1 Agent based Algorithm

In the infrastructure proposed, each IoT device/service is associated with a cyber agent that represents it in the cyber layer. The cyber agents perform autonomously, in a self organizing fashion and exploiting only local information, all steps of the algorithm in order to obtain an overlay network useful for recommendation operations. The topology of the emerging overlay depends of the algorithm strategy. In fact, on the basis of the selection policy of the neighbors of each cyber agent, it will emerge a given topology. In this case, the cyber agents execute a set of steps in order to obtain an ordered list on the basis of the similarity value with a given IoT device/service. The first element of the list will be the IoT device/service, among all, having the highest value of similarity with the wished IoT device/service, the second element is the cyber agent having the second highest value of the similarity, and so on. The steps performed by each cyber agent A_{cyber} having vector V_c in order to achieve the algorithm are reported in Algorithm 1. Here, H_{agents} list containing the linked cyber agents with vector value higher than Vc and L_{agents} list containing the linked cyber agents with vector value lower than Vc. The lists are computed through the function $computeList(H_{agents}, L_{agents}$. The function *identifyMaxSubmax*(H_{agents}) returns the linked cyber agents with the vector value having the maximum and sub-maximum similarity value with the current cyber agent, while the function *identifyMinSubmin*(L_{agents}), provides the linked cyber agents with the vector value having the minimum and sub-minimum similarity value with the current cyber agent. The function *createLink*(ca_a, ca_b) generates a virtual link between cyber agent ca_a and cyber agent ca_b. In detail, to ca_a is notified to add ca_b in its neighbors list and to ca_b is notified to add ca_a in its neighbors list. The removal of the cyber agent ca as neighbor of the current cyber agent is performed by means of function *remove(list,* Algorithm 1: Steps of the algorithm.

```
while true do
    computeList(H_agents, L_agents);
    if H_agents.length() > 1 then
        [max,submax]=identifyMaxSubmax(H_agents);
        createLink(max, submax);
        remove(H_agents, max);
    end
    else if L_agents.length() > 1 then
        [min,submin]=identifyMinSubmin(L_agents);
        createLink(min, submin);
        remove(L_agents, min);
    end
end
```

ca) which simply delete *ca* from the neighbors list.

When this starting phase finishes, the last cyber agent contained in the list L_{agents}, represents the linked cyber agent with the highest vector value among all linked cyber agents with the vector value lower than V_c; while, the H_{agents} list contains the linked cyber agent with the lowest vector value among all linked cyber agents with vector the value higher than A_{cyber}. At a steady situation i.e. after a transition phase, each cyber agent is connected, with virtual or real link, with two cyber agents: the cyber agent having the vector value immediately less similar and the cyber agent with the vector value of immediately more similar of the all cyber agents. It is possible to design a smart recommendation mechanism thanks to the ordering achieved by the algorithm. In fact, the cyber agents organize themselves based on the similarity with the wished IoT device/service and, once the algorithm ends, we can select a given number of IoT devices/service starting from the head of the list. Thanks to the organization, the suggestion provided are very similar and, more probably, useful for the user.

3 EXPERIMENTAL RESULTS

In order to investigate its effectiveness, a Java simulator was implemented in which the characteristics of real scenario are careful considered. The simulator allows to build a set of IoT devices/services randomly connected to a given mean number of others IoT devices/services (neighbors), in order to create a group of objects connected through a local area network. Each IoT device/services is described thorough a metadata: a text file containing a detailed description of the characteristics of the objects. By exploiting the *Doc2Vec* library, a cyber agent, responsible of an IoT service/device, can obtain a vector representing it. The cyber agents following autonomously

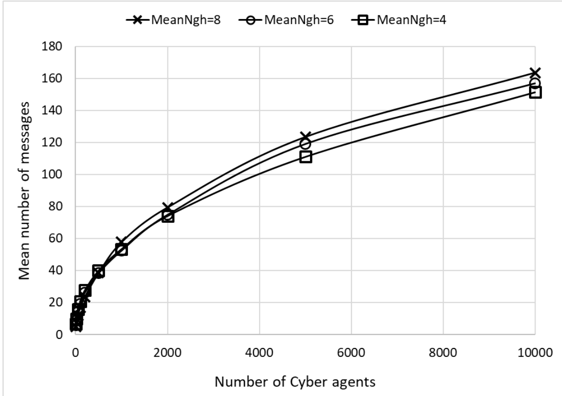

Figure 2: The average number of messages managed by each cyber agent to obtain the logical sorting, for different value of mean number of linked neighbors.

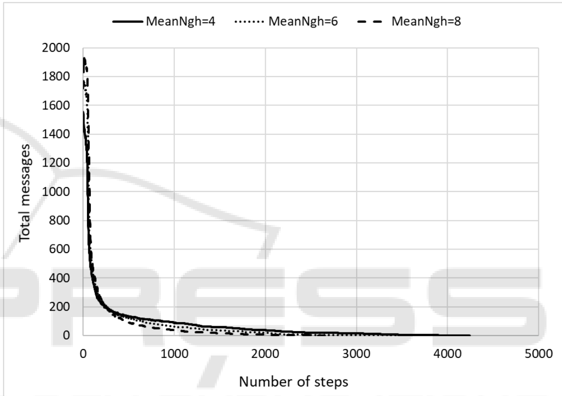

Figure 3: The total number of messages exchanged by all cyber agents per step to obtain the sorting. The number of involved cyber agents is set to 5000.

the steps of the algorithm organize in order to obtain the overlay useful to improve recommendation operations. Figure 2 shows the mean number of message managed by each cyber agent, for different number of IoT devices/services. The experiments were executed for different values of *MeanNgh*, that is the average number of connections/neighbors of every cyber agent.

It can be noticed how with a limited number of messages, the algorithm, reaches a stable situation and an useful overlay is obtained. In Figure 3 the total number of messages exchanged by all cyber agents per step to achieve the global organization, when the number of the cyber agents involved in the process in fixed to 5000, is shown.

Notice that the number of messages decreases exponentially and the algorithm converges in a finite number of steps. Successively the worst case to obtain the organization, i.e. the maximum possible number of steps needed to each cyber agent to individu-

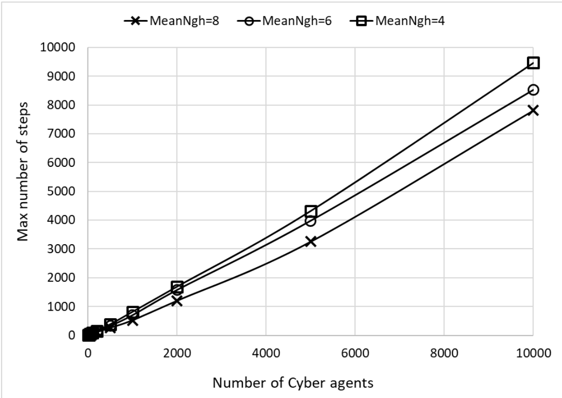

Figure 4: The maximum value of possible steps (worst case) necessary to obtain the logical sorting.

ate its neighbors, is shown in Figure 4. Even in this case, the experiments were executed for different values of the average number of connections/neighbors of cyber agent, when the number of the involved cyber agents changes. The interesting result is that the maximum number of steps needed to each cyber agent to individuate its neighbors is always very low and, anyway, limited to first application of the algorithm in which no one previous sorting exists.

4 CONCLUSIONS

The design of a multiagent platform for building a recommendation system in Internet of Things boasting semantic features, was presented. This platform relies on an agent based algorithm allowing discovery and recommendation operations faster. IoT objects are represented through vectors obtained by exploiting a word embedding library able to capture the semantic characteristics. Intelligent agents autonomously execute a distributed algorithm allowing to bring out an useful overlay for recommendation operations. Experimental results show as the algorithm, pillar of the platform proposed, enables an effective reorganization of IoT services/devices, allowing very encouraging performance for recommendation operations.

REFERENCES

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc.

Cai, M., Frank, M., Chen, J., and Szekely, P. (2004). Maan: A multi-attribute addressable network for grid information services. *Journal of Grid Computing*, 2(1):3–14.

Crespo, A. and Garcia-Molina, H. (2002). Routing indices for peer-to-peer systems. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 23–32. IEEE.

Forestiero, A. (2017). Multi-agent recommendation system in internet of things. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGrid '17, pages 772–775, Piscataway, NJ, USA. IEEE Press.

Forestiero, A., Mastroianni, C., and Spezzano, G. (2005). A multi agent approach for the construction of a peer-to-peer information system in grids. *Self-Organization and Autonomic Informatics (I)*, 135:220–236.

Forestiero, A., Mastroianni, C., and Spezzano, G. (2008a). Building a peer-to-peer information system in grids via self-organizing agents. *Journal of Grid Computing*, 6(2):125–140.

Forestiero, A., Mastroianni, C., and Spezzano, G. (2008b). Reorganization and discovery of grid information with epidemic tuning. *Future Generation Computer Systems*, 24(8):788–797.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339. Citeseer.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Mooney, R. J. and Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM.

Oppenheimer, D., Albrecht, J., Patterson, D., and Vahdat, A. (2005). Design and implementation tradeoffs for wide-area resource discovery. In *Proc. of the 14th IEEE International Symposium on High Performance Distributed Computing HPDC 2005*, Research Triangle Park, NC, USA.

Pazzani, M. and Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3):313–331.

Platzer, C. and Dustdar, S. (2005). A vector space search engine for web services. In *Web Services, 2005. ECOWS 2005. Third IEEE European Conference on*, pages 9–pp. IEEE.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM.

Selmi, A., Brahmi, Z., and Gammoudi, M. (2014). Multi-agent recommender system: State of the art. In *In Proc. of the 16th International Conference on Information and Communications Security*.