Natural language processing approach for distributed health data management

Agostino Forestiero
CNR - ICAR
Via Pietro Bucci, 8-9 C
87036 Rende (CS), Italy
mail: *agostino.forestiero@icar.cnr.it*

Giuseppe Papuzzo
CNR - ICAR
Via Pietro Bucci, 8-9 C
87036 Rende (CS), Italy
mail: *giuseppe.papuzzo@icar.cnr.it*

***Abstract*—Today's health domain is characterized by heterogeneous, numerous, highly dynamics and geographically distributed information sources. Moreover, the increasing use of digital health data, like electronic health records (EHRs), has led to store an unprecedented amount of information. Managing this large amount of data can, often, introduce issues of information overload, with potential negative consequences on clinical work, such as errors of omission, delays, and overall patient safety. Innovative techniques, approaches and infrastructures are needed to investigate data featured by high velocity, volume and variability. This paper introduces a distributed and self-organizing algorithm for building a management system for big data in highly dynamic environments like healthcare domain. Health data are represented with vectors obtained through the *Doc2Vec* model, a Natural Language Processing (NLP) approach able to capture the semantic context representing documents in dense vectors namely *word embeddings*. Doc2Vec is an unsupervised algorithm to generate vectors starting from sentences/documents based on *word2vec* approach which can generate vectors for words. The servers of a clinical distributed system, by performing autonomous and local operations, organize themselves in a sorted overlay network, so that resource management operations become faster and efficient. The effectiveness of the approach was proved performing a set of preliminary experiments exploiting a tailored implemented simulator.**

***Index Terms*—Electronic Health Records, Natural language processing, Semantic overlay network, Self-organization**

I. Introduction

Nowadays, most of data are generated of heterogeneous and geographically distributed infrastructures, where the information sources are extremely dynamics. In health domain, the increasingly utilization of digital infrastructures and data, like electronic health records (EHRs), has contributed to generate an unprecedented amount of stored information. Due to the information overload produced, several management issues arisen, with potential negative consequences on clinical work, such as errors of omission, delays, and overall patient safety [13]. In fact, managing large volumes of health data produced daily together with the effective handling of medical acts and processes are needed as well. Building an appropriate system to efficiently support all the requirements of the health environment, is one of a main today challenge. Mechanisms for searching useful patient's EHRs in shared health system were proposed, but most of them are based on a central approach [15], and then can be acceptably tackled with networks with limited size. In large and dynamic systems in fact, centralized mechanisms show limits and it is preferable to design innovative and distributed approaches. To organize and discovery resources in distributed systems, thanks to their inherent robustness and scalability, several peer to peer approaches [5]–[8], [11], were proposed. The objective of these approaches is to allow users to locate resources or services, either hardware or software, with required characteristics. Usually in P2P systems, resources and services are represented through a syntactical/ontological description of their characteristics, *metadata*, and thus the discovery service has to locate the metadata containing the wished features. Often, metadata are indexed through vectors that can have two possible different meanings [3] [14]: (i) the presence or absence of a given *topic* can be represented of a bit; and (ii) vectors map metadata through a tailored function. Locality sensitive hashing (LSH), for example, is a technique used to realize an efficient approximate nearest-neighbor search. All similar metadata are mapped in near high-dimensional vectors with high probability than distant ones, by exploiting the locality sensitive hash function [4]. As result, similar vectors are assigned to resources with similar characteristics, i.e. having similar metadata [12]. In the approach proposed in this paper, the *Doc2Vec* model [9], able to represent documents in dense vectors, also capturing the semantic, is exploited to map metadata. *Doc2Vec* is an unsupervised NLP based algorithm able to generate vectors starting from sentences/documents based on *Word2Vec* approach which can generate vectors, namely *neural word embeddings*, for words. *Word2Vec* [9] 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. The "word embedding" approach, one of the most significant recent developments in natural language processing, is able to capture multiple different degrees of similarity between words. Natural Language Processing (NLP) [10] [1], is a branch of artificial intelligence (AI) that investigates the interaction between machines and humans using the natural language. To create the model of relationships between the words, a

2377-5750/20/$31.00 ©2020 IEEE
DOI 10.1109/PDP50117.2020.00061

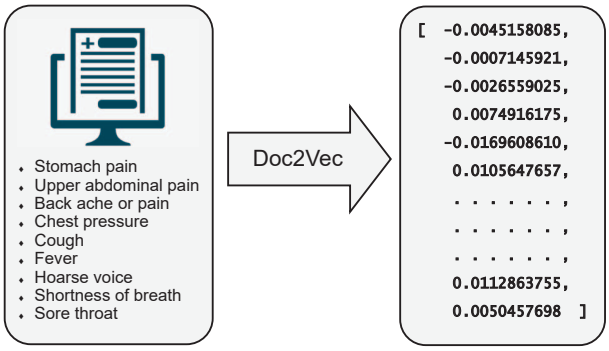

Fig. 1. An example of how the *Doc2Vec* library works.

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 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.

II. A DISTRIBUTED SORTING ALGORITHM EXPLOITING EHR SIMILARITY

The aim of the algorithm is to builds an information system through a logically organized overlay of clinical servers in order to improve discovery operations of useful information. Exploiting a self-organizing approach, each clinical server executes in completely autonomous mode a set of simple operations, based on local information and, at global level, emerges a sorted overlay. The servers contain several kind of clinical data, like EHRs, which can be represented through vectors obtained by means of a *Doc2Vec* library. In this way, it is possible to achieve a logical sorting among servers by defining a metric based on their vectors. The distance/similarity between two clinical server can be computed through the cosine distance/similarity between the vectors representing the data. Given two electronic health records, the cosine measure utilized to compute the similarity between them is reported in formula (1). Here EHR_1 and EHR_2 indicate the vectors obtained through the *Doc2Vec* library.

$$cos(\overrightarrow{EHR_1}, \overrightarrow{EHR_2}) = \frac{\overrightarrow{EHR_1} \cdot \overrightarrow{EHR_2}}{|\overrightarrow{EHR_1}| \times |\overrightarrow{EHR_2}|} \quad (1)$$

The outcome is an overlay of sorted servers: each clinical server has a logical link towards two servers, one with vector value lower and one with the vector value higher. The server with the vector with the absolute minimum value and the absolute maximum value of all vectors in the network, will be linked to a unique server. The main steps performed by each server $server_c$ with vector $vector_c$ in order to achieve the algorithm are:

- compute L_s and H_s list containing the linked servers with vector value lower and higher than $vector_c$, respectively;
- if the length of L_s/H_s list is greater than 1, identify the server with the minimum/maximum and the server with the sub-minimum/sub-maximum vector value;
- notify through messages the servers with the minimum/maximum and sub-minimum/sub-maximum vector value with the information related to the new virtual neighbor - a virtual link can be created between the two servers; server receiving a message, updates its L_s/H_s list to considerate this new neighbor in the successive computation.
- remove the server with the minimum/maximum vector value in L_s/H_s list.

The last server contained in the list L_s is the linked server with the *highest* vector value among all linked servers with the vector value lower than $vector_c$; while, the H_s list contains the linked server with the *lowest* vector value among all linked servers with vector the value higher than $server_c$. At a steady situation i.e. after a transition phase, each server is connected with two servers (through a real or virtual link): the server having the vector value immediately lower and the server with the vector value of immediately higher of the overall network. A sorted overlay of linked servers emerges that allows an informed and faster discovery procedure. The infrastructure autonomously adapts itself to joins and departs of servers and to the variation of the characteristics of the data contained in the server. When a clinical server leaves the system, it simply has to notify to its neighbors the information each other, in this way they can link between them. When a new server joins the system, it simply has to communicate the arrival to the neighbors by exchanging the vector with them, and then simply to execute the steps of the algorithm. It will be involved in the logical reorganization and properly entered in the overlay.

The algorithm proposed in this paper is a basic component of a framework able to collect and analyze huge volumes of heterogeneous clinical data in order to automatize disease diagnoses [2]. Data involved in the process can be originated by different and distributed sources. Medical data like electronic health records, laboratory test data, symptoms of patients, electrocardiography (ECG) results and demographics to identify similar patients, patient health logs or medical images, social networks data, and data coming form wearable devices, are some examples of useful health data sources. A set of innovative artificial intelligence methods exploit such data for recommending possible diseases to physicians. Several services ranging from acquiring data from disparate sources (e.g., sensors, smart phones, ECG, etc) to the integration analysis and processing of such data in order to define a set of services for disease diagnoses and treatment, are designed and implemented.

The architecture of the framework is envisioned into three layers. The first layer allows for the collection, integration and handling of different categories of large-scale, fragmented health medical data produced from disparate sources. The goal of the artificial intelligence (AI) layer is to apply machine learning and deep learning approaches to build models that allow an automatically generated context-based and rich representation of health-related information. The AI layer is composed of a set of specialized modules, like Natural language Processing (NLP) module, radiomics module, predictive analytics, providing physicians with specific knowledge, patients' information and intelligent applications, which can improve the efficiency of their decision-making processes. In particular, a key application offered by the framework is categorization of text fragments, at a sentence level, based on the emergent semantics extracted from a corpus of medical text by exploiting convolutional neural networks (CNNs) that have greatly improved traditional machine learning approaches, since they have the ability to learn complex feature representations. The third level makes available a CPU/GPU infrastructure that allows framework to handle large amounts of data in a limited processing time.

A. *Discovery procedure exploiting the sorted overlay network*

It is possible to design a smart search mechanism thanks to the ordering achieved by the algorithm. The idea is: at every step, queries are issued towards clinical servers with vector as more as similar to the target content. In this way, an *informed* search mechanism obtained thanks to the overlay, can be exploited. The approach is fully distributed and the messages are forwarded only on the basis of local information, i.e. evaluating the similarity of the target vector and the vectors of the neighbors. At each step, the query can be simply issued towards the "best neighbor clinical server", thanks to the organization performed by the algorithm. The linked server, both belonging to the emerged overlay or to the real network, with the minimum distance value from the target vector, is the best neighbor clinical server. The informed procedure guarantees that, at each step, the search is always more close to the vector having the highest similarity with the target vector.

The discovery procedure ends when none vector of the neighbor servers is more similar than the vector of the local server, therefore the similarity value cannot be further improved. A message can be forwarded towards the clinical server (user) with information related to the address of the clinical server with highest content similarity. Alternatively, queries during their journey can collect information related to clinical servers with a vectors exceeding a given threshold of similarity and provide a list of results among which the user can choose.

III. Experimental results

The effectiveness of the algorithm was proved implementing a Java simulator in which the characteristics of real networks were careful considered. The simulator allows to easily execute an extensive set of simulations exploiting a visual interface,

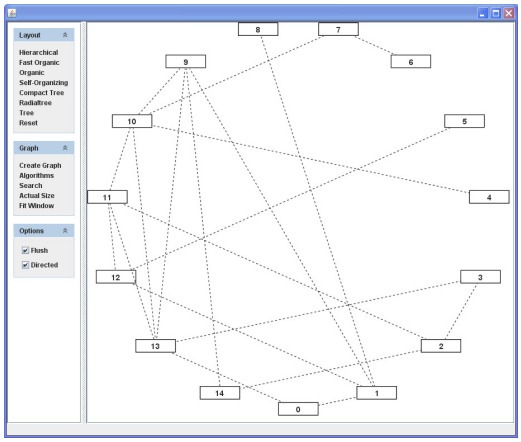

Fig. 2. An example of clinical server network.

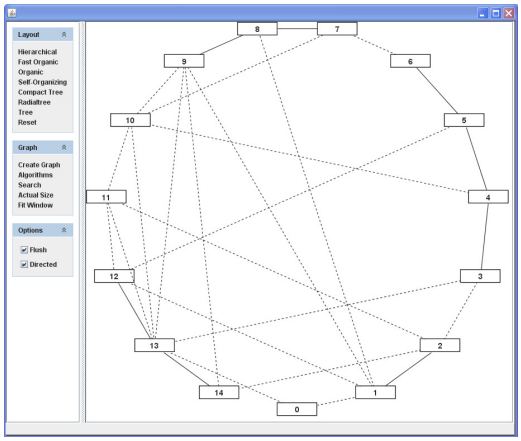

Fig. 3. The outcome of the simulator after the algorithm.

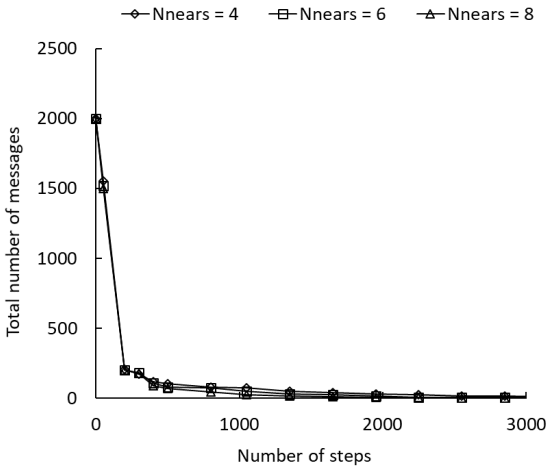

Fig. 4. The traffic per step (overall number of messages exchanged per step) in the network to obtain the organization with 5000 clinical servers.

that provides the possibility to vary the algorithm parameters and immediately evaluate the results through a graphical outcome. Figure 2 reports, as example, a snapshot of the network situation before the algorithm utilization. In Figure 3, the overlay emerged after the utilization of the algorithm, is depicted. For the sake of simplicity, in the figure, the servers are represented with a number. Figure 4 reports the total number of messages exchanged by all servers per each step to obtain a stable situation. The network size is set to 5000. Notice that the algorithm converges in a finite number of steps and the number of messages decreases exponentially. A large number of messages is necessary to reach the sorting only for the initial running because none previous order exists and the system is completely disordered. In Figure 5, the solid

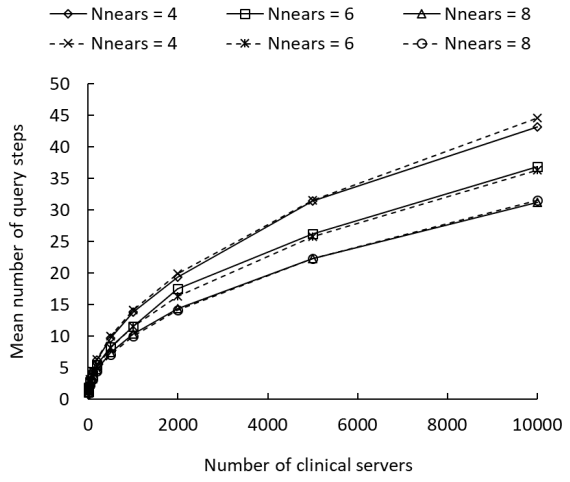

Fig. 5. The mean number of query hops needed to locate the target content. Dotted lines depict the mathematical results, while the solid lines report the simulation results.

lines depict the mean number of hops, N_σ, performed by a search query to locate the server in which the target is located. The experiments show the results for different number of clinical servers involved and for different mean number of *real* neighbors for each server. Notice that, a low number of hops is sufficient to locate the target server even when the network size is very large. Moreover, figure 5 reports a comparison among the results obtained through simulations (solid lines) and a mathematical evaluation, reported in formula 2, empirically derived performing an extensive set of experiments (dotted lines).

$$N_\sigma = \kappa * \sqrt{\frac{N_{servers}}{N_{nears}}} \tag{2}$$

The parameter κ is equals to 0,9. We can remark how both values are very close. $N_{servers}$ is the number of clinical servers involved. Notice that, the mean number of query hops, N_σ, is proportional to the square root of the ratio between the number of clinical server involved and the mean number of neighbors for each one, N_{nears}.

IV. Conclusions

This paper introduced a distributed and self-organizing NPL based algorithm for building a data management system in highly dynamic environments like health domain. Each health data, like electronic health record (EHR), contained in a clinical server, was indexed through a semantic vector obtained through the artificial neural network model *Doc2Vec*. Each server performs autonomous and local operations in order to obtain a sorted overlay network of clinical servers. Thus, user requests can efficiently reach the server containing the target content or service through an *informed path*. Experimental results showed the effectiveness of the approach, how the discovery operations became faster and are proportional to the square root of the network size.

References

[1] Collobert R., Weston J.: A uni-fied architecture for natural language processing: deep neural networks with multitask learning. In Proc. of ICML, pp 160–167 (2008)

[2] Comito, C., Forestiero, A., Papuzzo, G.: Exploiting Social Media to Enhance Clinical Decision Support. In Proc. of IEEE/WIC/ACM International Conference on Web Intelligence. Thessaloniki, Greece, pp. 244–249 (2019)

[3] Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. In: Proc. of the 22nd International Conference on Distributed Computing Systems ICDCS'02. pp. 23–33 (2002)

[4] Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In Proc. of the Twentieth Annual Symposium on Computational Geometry. pp. 253–262. SCG '04, ACM, New York, NY, USA (2004)

[5] Folino, G., Forestiero, A., Spezzano, G.: A jxta based asynchronous peer-to-peer implementation of genetic programming. Journal of Software **1**(2), pp. 12–23 (2006)

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

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

[8] Forestiero, A., "Multi-Agent Recommendation System in Internet of Things" 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid, pp. 772–775 (2017)

[9] Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In Proc. of the 31st International Conference on International Conference on Machine Learning - Volume 32. pp. II–1188–II–1196. ICML'14 (2014)

[10] Liddy E.: Natural language processing, 2nd edn. Encyclopedia of Library and Information Science, Marcel Decker (2001)

[11] Nobre, J.C., Melchiors, C., Marquezan, C.C., Tarouco, L.M.R., Granville, L.Z.: A survey on the use of p2p technology for network management. Journal of Network and Systems Management pp. 1–33 (2017)

[12] Oppenheimer, D., Albrecht, J., Patterson, D., Vahdat, A.: 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 (July 2005)

[13] Pivovarov, R., Elhadad, N.: Automated methods for the summarization of electronic health records. Journal of the American Medical Informatics Association **22**(5), 938–947 (2015)

[14] Platzer, C., Dustdar, S.: A vector space search engine for web services. In Proc. of the Third European Conference on Web Services ECOWS 2005. p. 62. IEEE Computer Society, Washington, DC, USA (2005)

[15] Pruski, C., Wisniewski, F.: Efficient medical information retrieval in encrypted electronic health records. Quality of Life Through Quality of Information: Proc. of MIE2012 **180**, 225–229 (2012)