JVLC

Journal of Visual Language and Computing

Volume 2023, Number 1

DOI: 10.18293/JVLC2023-N1

Journal preparation, editing and printing are sponsored by KSI Research Inc.

Journal of
Visual Language and Computing

An Open Access Journal published by

KSI Research Inc.

156 Park Square Lane, Pittsburgh, PA 15238 USA

JVLC Editorial Board

Journal of
Visual Language and Computing

Volume 2023, Number 1

August 2023

Table of Contents

Regular Papers

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Decision Support Enhancement Through Big Data Visual Analytics

Pierfrancesco Bellini[a], Enrico Collini[a], Paolo Nesi[a,*], Alessandro Luciano Ipsaro Palesi[a] and Gianni Pantaleo[a]

[a]DISIT Lab, Department of Information Engineering, University of Florence, Italy

ARTICLE INFO

Article History:
Submitted 2.9.2023
Revised 3.20.2023
Accepted 4.2.2023

Keywords:
Decision Support
Big Data
Visual Analytics
Internet of Things

ABSTRACT

The development and diffusion of Internet of Things, Artificial Intelligence, Business Intelligence and Visual Analytics solutions has brought new paradigms to exploit Big Data analytics into Decision Support Systems. The increasing requirements of automating Decision Support Systems and making them more efficient and reliable represent a research field which has recently attracted a lot of interest and efforts. The main challenges are represented by the initial black-box nature of machine learning and deep learning methods, which often cannot deal with the necessity to provide explainable results, which is a critical aspect when dealing, for instance, with automated decision processes in legal and administrative contexts. This paper presents a study of the main concepts and requirements for enhancing Decision Support through big data visual analytics, presenting the Snap4City solution and describing some real use cases in which it has been and is currently being exploited.

1. Introduction

The recent advances in ICT related to Big Data, Internet of Things (IoT) and Artificial Intelligence (AI) solutions have provoked a great interest in investigating the possibility to enhance and improve automatic decision-making processes. The enhancement and automation of Decision Support Systems (DSS) are more and more required in several different contexts (smart city and related domains, industry 4.0, etc. [1]) for a wide range of scenarios and use cases (e.g., public administrations and municipalities, private companies, and regular citizens).

The implementation of DSS based on a model-driven approach has to face the high complexity of many different real use case decision problems, which consequently rely in a large variety of different models. Therefore, it can be difficult to provide suitable and reliable models supporting the automation of the specific decisional process.

*Corresponding author:
Email address: paolo.nesi@unifi.it
Website: https://www.disit.org/

Data-driven decision-making is the process of using evidence and insights derived/learnt from data to guide the decision-making process and to verify a plan of actions before it is committed. In the era of Big Data, this approach is being more and more frequently adopted, and it has been observed that the confluence of data analytics with Big Data can significantly improve the way DSS can be designed and implemented, and how this approach can impact on a company's performance [2]. Big Data Analytics is a way to refer a comprehensive approach to process and analyze big data sets by applying advanced analytics techniques, improving data-driven decision making and support [3]. However, data-driven decisions are still prone to errors and failures, as addressed in [4], in which it is discussed the case of decisions addressed for COVID-19 forecasting, which have often failed, due to several factors such as the small number of input data, lack of historical data as reference. Actually, the quality of decisions depends not only on data, but also on the way in which the data is collected and processed [5]. Furthermore, the automation of data-driven decisions or even the automated suggestions still remains debatable, and there are many examples in literature about issues and causes of failures of data-driven automated decision

DOI reference number: 10-18293/JVLC2023-N1-026

making and supports [6].

In [7], it is highlighted how incorrect decisions, generated/suggested by the automated decision systems in the Swedish public employment service, enforce the evidence that human users have to cover an active role in the decision-making process. In [8], mistakes are reported in the automated Finnish tax assessment process, and this remarks how Explainable AI techniques and methods (e.g., explainability and accountability) are important and should be considered for automated decisions in legal/administrative contexts. In [9], the lack of reliable and accurate evidence in data-driven decisional process at Royal Dutch Shell Telecommunications company led to many problematic decisions. This implies that the quality and amount of data affect the decisions. In addition to this, the same study reports that human decision makers often received more information than they could process, addressing the aspect that human decision makers have a limited capacity for processing data on their own, and therefore DSS are greatly required. Authors in [10] present the case of Danish Primera Air, where the inability of the company to capture big data and use Big Data Analytics for strategic decisions led to the company failure. This event represented a strong motivation towards the adoption of efficient and reliable Big Data platforms.

In this paper, the evolution of DSS and their application in the context of big data platforms for smart city is reported. The paper is focused on the main evolution performed in Snap4City platform to support the design and the implementation of smart applications and DSS. Snap4City is an open-source platform deployed and used in several smart city and industry 4.0 installations. The biggest installation of the framework is a multi-tenant platform, managing advanced Smart City IoT/IoE applications with 20 organizations, 40 cities and thousands of operators and developers. Snap4City is an official FIWARE platform and solutions, it is compliant with security and privacy aspects [18], [20], and provide support for the development of a large range of ML and AI solutions, including what-if analysis and DSS.

The paper is structured as described in the following: in Section 2, the main concepts and requirements for building efficient AI-based Big Data Analytic solutions for decision support are reported. Section 3 presents the proposed solution, the Snap4City framework, which include an ML/AI enabled Big Data Analytic platform for decision support. Section 4 reports some real use cases implemented in the Snap4City platform. Finally, Section 5 is left to conclusions.

2. AI-based Big Data Analytics: Concepts and Requirements

Descriptive, prescriptive and predictive solutions have been provided since many years, from statistic, operating research, and regressive models. In most cases, they have not been capable to provide satisfactory results for their limitation of discovering/modelling complex functions and relationships. Early Machine Learning (ML) and Deep Learning (DL) solutions mainly had a black box nature which needed explainability and interpretability before their effective application in real-world cases and critical situations [11]. On the other hand, ethical aspects (on data and processes) are very sensitive and a wrong assumption in taking data and/or setting up solutions may lead to biased results/suggestions, which may correspond unfairness, discriminations, and this may lead to unforeseen costs [12]. The IoT (Internet of Things) combined with Big Data are enabling a large number of new data analytics. Big Data Analytic with AI play a strong role in leveraging businesses and solutions providing reliable predictions, prescriptions, early warning, classifications, detections, suggestions, etc., thus enhancing automated/semi-automated decision support processes. Actually, these technologies can lead to reduce costs and increase the efficiency of business and production processes. This also implies to add value to collected Big Data, extracting from them new knowledge, hints, strategies, mitigations, and discovering information and implications never detected before.

2.1 Context and Application Scenarios

The applications of the above-described aspects can be experienced in almost any domain of smart city and industry: mobility, health, energy, environment, waste, chemistry, manufactory, delivering, agriculture, etc. The resulting advantages can be for final users, for decision makers and thus for city/companies. For example, the efficient parking prediction models and tools (smart mobility) [13] allow to reduce the social cost of parking search, in terms of reduction of fuel consumption, producing less NO2 and other pollutants. Optimization of services such as waste collection (smart environment), which implies a cost reduction by reducing the number of trucks/trips needed for waste collection. This is an advantage for the quality of life of city users, and a reduction of costs for the administrators. A second example is related to the predictive maintenance in smart industry, Industry 4.0 [14] for reducing the costs for intervention, due to unexpected faults and stops of the services/productions. Predictive maintenance also implies a further reduction of the production costs by improving company's efficiency and resilience. A third example is related to assess and predict reputation of services (for example: mobility services, restaurants, museum) from social media to increase the quality of services, producing suggestions and thus to reduce pikes in the demand and promote alternative offers and solutions.

Moreover, one of the main common goals of DSS may be to prepare cities and industries to be more resilient to the so called *unexpected unknown* events, natural or

provoked disasters, by integrating simulations and ML/AI solutions, enabling what-if analysis in quasi real time, and increasing resilience and capacity. Reacting to unexpected unknown events in faster manner leads to the reduction of the recovery costs, thus mitigating the overall risks and damages.

The most relevant challenges for cities in the coming years include the energy saving and the ecological transitions KPI (Key Performance Indicators), the Sustainable Development Goals (SDG) [15] (reported in Figure 1), promoting more livable cities according to 15 Min City Indexes [24] and Driving Urban Transitions, DUT, to a sustainable future.

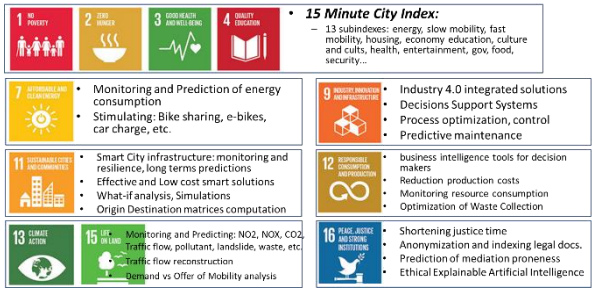


Figure 1: Sustainable Development Goals, a part of them more related to Snap4City Smart City solutions.

The Snap4City platform is an open source framework, developed by the DISIT Lab of the University of Florence (as described in more details in Section 3), aiming at satisfying all the requirements and features described in the following.

2.2 Big Data Analytics and Explainable AI

Big Data Analytics in the last few years rapidly evolved and started to be used in complex systems and not only to make direct predictions and prescriptions.

An efficient framework for supporting decision makers through AI-based Big Data analytic solutions should provide the capability to exploit AI techniques (including: short-term and long-term forecasting and classification models, suggestion and recommendation systems, etc.) to perform system modelling and simulation (based on the different applicative scenarios). The exploitation of these techniques can be used to create solutions for Early warning and what-if analysis. They can be used to define resilience countermeasures, disaster recovery and to build more informed strategies, mitigations and plans. A conceptual overview of these topics is illustrated in **Figure 2**. In this view, a relevant role is played by the formalization of the scenarios, and in the identification of the DSS targets in terms of quality, KPI, SDG, etc.

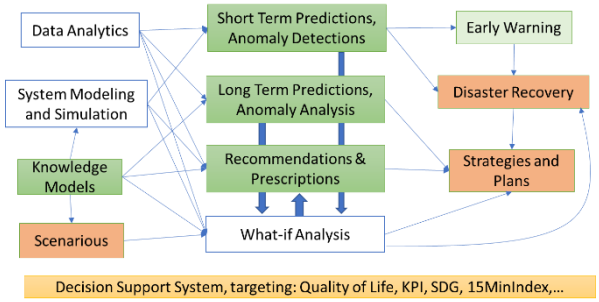


Figure 2: Data Analytics based on AI solutions for decision support systems.

Therefore, Decision Makers started to use the new solutions with the expectation and the strong need of respecting ethics on data and processes. To this end, ML/AI trustworthiness, Data Ethics and AI Ethics approaches have been studied and applied in the context of decision makers [16]. Data Ethics refers to the aspects that may provoke a bias and ethical problems since the training phase. For example, training the AI with biased data, unbalanced distribution of cases, etc. Moreover, specific ML/AI methodologies and solutions for Explainable Artificial Intelligence (XAI), are presently providing support in this direction [17], since they are capable to explain the rationales behind the typical results provided (global explainable AI) and may provide specific description/rational for each result/suggestion provided (local explainable AI). XAI typically adds value to the provided decision producing hints and discovering implications and correlations never detected before. Therefore, as in Snap4City, such an integrated approach including ML/AI trustworthy, Data Ethics and AI Ethics must be enforced into the development process and life-cycle (as depicted in **Figure 3**) of the new smart platforms and solutions.

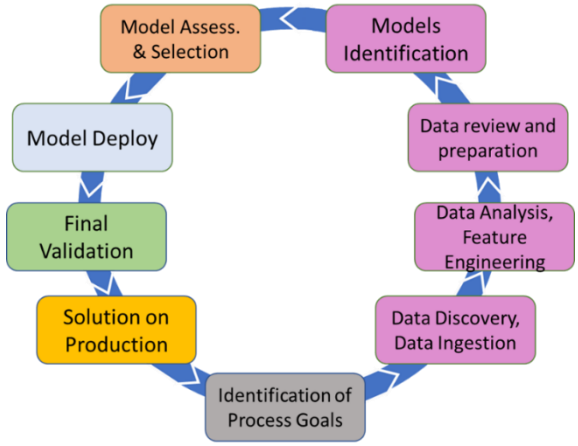


Figure 3: Development life cycle of AI-based smart solutions: adopted/suggested by Snap4City platform.

2.3 Legal Aspects and Privacy

Any ML/AI/XAI solution (including data ingestion, transformation, training, visualization, etc.) has to

respect data privacy policies. i.e., being compliant with GDPR [18], [20] (General Data Protection Regulation of the European Commission) and/or similarly regulations in other non-European Countries (e.g., the California Consumer Privacy, CCPA). These aspects have to be addressed since the beginning of the above presented life cycle (see **Figure 3**), when the data discovery and data ingestion are performed, and in particular in the data analysis phase. This also means that the solutions have to respect the Data sovereignty, for which the data are subject to the laws and governance structures of the nation/country where they were collected/produced. Specific licenses can be modelled on Snap4City and related development tools enable the development of AI, while each single implementation has to guarantee the respect of Data sovereignty and GDPR. One of the main problems on AI may be seen as the need of accessing to specific data related to the person behavior, or collective behavior. On the other hand, the state of the art literature and the long track in data analytics also demonstrated that in a large number of cases, surrogated data can be found to substitute those that are protected or too private to be used (in some cases, at the expenses of a small reduction in precision). Specific techniques for anonymization preserving static validity may help in this sense and are getting a larger diffusion.

2.4 Requirements

On the basis of the analysis conducted in the several developments of Snap4City solutions, a list of requirements for enabling AI-based Big Data analytic solutions in decision support systems is proposed. Specifically, In the context of IoT Enabled Smart Cities and Industry 4.0, the European commission has fostered and stimulated the analysis of this kind of requirements, for instance: EIP-SCC (https://eu-smartcities.eu/), the European Innovation Partnership on Smart Cities and Communities; ENOLL (https://www.openlivinglabs.eu/), the European Network of Living Lab, supporting real-life test-bed and experimentation environments. Moreover, the Select4Cities (https://www.select4cities.eu/) consortium was one of the largest pre-commercial procurement, involving the cities of Copenhagen, Antwerp and Helsinki, created to identify the best solutions putting together the smart city context and the living lab and all the defined requirements. Snap4City has been the winner of the Select4Cities PCP in the 2019.

Requirements of IoT-enabled contexts, use cases and scenarios must take into account the complexity due to the increasing number and types of data sources, IoT protocols and brokers. The goal is to aggregate all these kinds of heterogeneous data into well integrated representations and tools for decision support to be provided in some interactive dashboards and visual analytic tools. To this aim, a first important step is represented by data collection and storage, which has to be compliant with all the afore mentioned protocols and formats. Also, semantic storage and triplification is required, to infer and build additional knowledge. Business intelligence tools are important to enable and enhance decision support. Business intelligence may request some data analytics, for example to calculate alternative routes after closing traffic for example in the areas of the city where air quality sensors are giving too high values for pollutant concentration, or for example in the context of smart parking to forecast free slots, for smart waste management to optimize the routes for collecting waste for trucks, etc. Finally, the platform has to provide complete visualization tools. Visualization should be interactive, to send commands from the user interface to the processing back-end, selecting some data transformation to build business intelligence tools.

In the following, a list of the identified requirements for enabling AI-based Big Data analytic solutions in decision support systems is provided:

- **Exploit AI-based Big Data analytics**, as discussed in the previous sub-sections.
- **Provide visual interfaces for users**, e.g., city dashboards, visual applications etc., in order to offer smarter and more accessible services to final users.
- **Data-Driven and Event-Driven approach by enabling IoT solutions**, to have the possibility to implement data-driven applications through the web without the necessity to install local applications, exploiting the development of IoT (the Internet of Things), IoT Edge.
- **Collect historical data** and provide access through API and/or microservices. It is also necessary to build a number of knowledge bases and semantic ontologies that can be queried and navigated, in order to understand what is going on in a specific domain.
- **Serve as Living Lab**, in order to foster open innovation, collaborative work, sharing of data, processes, visualization tools, experiences, solutions. The management of decision-making processes always involve a community of users, organizations and stakeholders.

In addition, also the following non-functional requirements have been identified:

- **Open Source**, following the Open Standard for communication and API.
- **Interoperability** regarding different kinds of protocols, formats, internal and/or external API, capable, with the possibility to be open to proprietary protocols as well.
- **Scalability and Robustness**: the architecture should be distributed and decoupled, modular, microservice-oriented.
- **Security by Design**: compliance to HTTPS, TLS standards and global/local regulations.
- **Privacy by Design (User Centric Design)**: compliance to GDPR and other global/local

regulations for data privacy and personal data management.

3. The Snap4City framework

In this section, the Snap4City solution (https://www.snap4city.org/) is presented, with the aim of responding to the requirements previously defined in Section 2. The Snap4City platform allows to collect data of any kind (dealing with most file formats and protocols) to save them into a Big Data store in which they can be queried for recovering specific historical data. The same storage can be used to collect data in real time and to save data analytic results. An overview of the Snap4City architecture is shown in **Figure 4**.

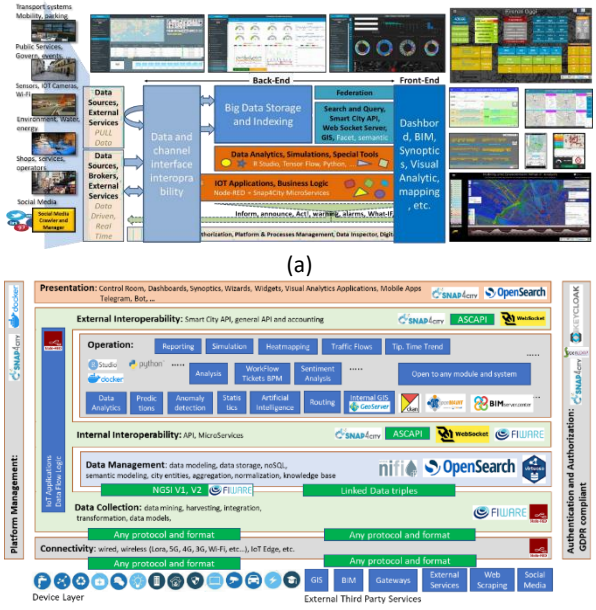


Figure 4: Overview of the Snap4City functional (a) and technical (b) architecture.

The general workflow includes the following activities:

- Big Data ingestion (historical and real time data collection and update, data transformation, see the Data Collection and Data Management layers in Figure 4b).
- Big Data Analytics AI/XAI tools, including Data transformation and dataset construction for implementing ML/DL predictive models training and validation (see the Operation Layer in Figure 4b).
- Model execution, taking in input the real time data, and the Model Fit to produce predictions which could be estimated 24 hour in advance and may be used to inform the civil protection, municipality, etc. The resulting model assesses in real time the probability of landslide events as early warning/prediction.
- Visualization of data and results (see the Presentation layer in Figure 4b) by means of visual analytic tools such as Dashboards (exploiting a large variety of graphical widgets), Mobile Apps, etc. Visualization should be interactive, to send commands from the user interface to the processing back-end, selecting some data transformation to build advanced business intelligence tools.

In Snap4City, many different activities of the previously described workflow, such as data ingestion and data analytic processes, are performed by using Node-RED applications on docker containers. The Node-RED visual language and environment allow to create IoT enabled application flows (see the IoT Applications Layer in Figure 4b) that can exploit the platform MicroServices with a specific node.js library [19]. In this way, users can build their own Business Logic supporting data-driven/event-driven paradigms. Data Analytics processes can be developed by using Python and/or Rstudio and be executed through dedicated Node-RED IoT Applications. IoT Apps may also allow to send alerts via Telegrams, SMS, emails, for alerting based on Data Analytics results.

Considering the requirements presented in Section 2.4, Snap4City enables AI-based Big Data Analytics, respecting ethics and GDPR compliant [20]. Snap4City has developed a large number of solutions in the context of Smart City and Industry 4.0 [21], [22], [23], serving as Living Lab by supporting users, developers and organizations to act in the platform at different levels. Snap4City fully supports the development of real time data analytic processes through trustworthy XAI. Snap4City is distributing a number of Open-Source data analytics tools and algorithms for: prediction, anomaly detection, classification, detection, constrained routing, optimization, analysis of demand vs offer of transportation. Data Analytics is fully integrated into What-IF analysis tools in control rooms, defining scenarios and solutions for operators and users.

4. Real Use Cases

The Snap4City framework is largely employed in many real uses cases and scenarios, in a wide number of smart cities, municipalities, industry 4.0 and companies in many different countries in Europe. The solution is applied in several different domains, such as mobility, industry 4.0, tourism management, smart waste and environment, 3D city digital twin representation. In the following, some real uses cases are described.

A data analytics and predictive model use for landslide forecasting has been presented in [21]. In this use case, the Snap4City platform has been used to ingest information from IoT Sensors and Open Data on the kind of terrain, the slope, the amount of cumulated rain, humidity in the Florence metropolitan area (Italy). Several predictive models for early warning have been designed, implemented and applied to collected data features: Random Forest (RF), eXtreme Gradient

Boosting (XGBoost), Convolutional Neural Networks (CNN) and Autoencoders different models have been assessed, and the XGBoost model resulted to be the best in terms of MAE, MSE and RMSE. An overview of this use case is illustrated in **Figure 5**.

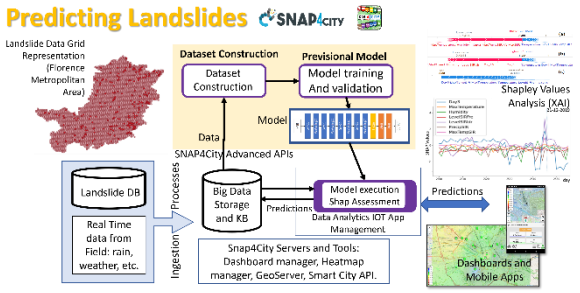


Figure 5: Landslide prediction framework exploiting the Snap4City platform.

Explainable AI tools have then been used to explain why a certain level of risk has been estimated by the predictive model, and which feature has a higher correlation with the result of the prediction. In the specific, Shapley values have been used, which measure the average expected contribution of each single feature with respect to all possible feature combinations. In particular, the features that resulted to contribute most to the prediction of a landslide event are precipitation level, temperature and phreatimetric data.

Another real use case is related to Industry 4.0. In [14], a Deep Learning solution for short term prediction of the working status of the Altair chemical plant has been presented. A DL model based on Long Short Term Memory Neural Networks (LSTM) and CNN has been used to provide one hour prediction of the plant status and indications on the areas in which the intervention should be performed by using explainable AI techniques (i.e., Shapley values showing that the most relevant features in fault determination were the sodium hypochlorite levels and Potable Ferric rate in two different lines of the plant). All data and results are shown in a maintenance dashboard (see **Figure 6**) in which it is possible to see the number of maintenance events in a chosen time period: the average and median of the number of hours needed to complete a maintenance intervention (intended as the difference between the start and end of intervention datetime).

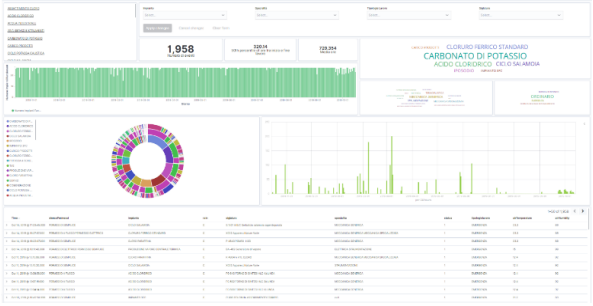


Figure 6: Maintenace Dashboard for Industry 4.0 Business Intelligence.

In [23] a method based on Bidirectional Long Short-Term Memory networks (Bi-LSTM) has been employed to provide predictions of available bikes in bike sharing racks, even with a limited amount of historical data. The solution was validated by using data collected in bike-stations in the cities of Siena and Pisa (Italy). In addition, an analysis of features relevance based on SHAP that demonstrated the validity of the model for different city cluster behaviors.

5. Conclusions

In this paper, a study and analysis of concepts to enhance decision support systems (DSS) through big data visual analytics have been presented. The main results that have been found by reviewing the state of the art, report that model-driven approaches for implementing DSS cannot properly handle the complexity of the many different real-world decision problems it must address. However, relying on data-driven decisions only can also lead to errors and failures, as reported in many use cases described in literature. To improve the effectiveness of DSS and support the Sustainable Development Goals, it is necessary to improve the improve the authority of AI-based decision support, in order to trust and increase the automation level of decision-making processes. To this aim, requirements have been provided in Section 2 of the paper, with a particular focus on discussing aspects such as the integration of Explainable AI in Big Data analytics, the management of legal aspects and privacy issues. In the specific, an efficient framework supporting decision makers through AI-based Big Data analytic solutions should provide the capability to exploit AI techniques to perform system modelling and simulation, easily adapting to different applicative scenarios. This can lead to smarter solutions for Early warning and what-if analysis, improving the environment resiliency, the adoption of countermeasures etc. To this aim, the Snap4City framework has been presented in Section 3, as a proposed solution to address all these aspects. Actually, the platform provides the capabilities to ingest, store and transform Big Data; exploit Big Data Analytics AI/XAI tools for implementing and executing ML/DL predictive models training and validation; visualization of data and results through Dashboards and Mobile Apps. In Section 4, a number of use cases have been also described to validate the proposed solution in real world scenarios and contexts.

References

[1] Bellini, P., Nesi, P., Pantaleo, G. (2022) “IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies”. Applied Sciences, 12(3).DOI: https://doi.org/10.3390/app12031607

[2] Frisk, J.E., Bannister, F. (2017) “Improving the use of analytics and big data by changing the decision-making culture: A design approach”. Management Decision, 55(10), pp. 2074-2088. DOI: https://doi.org/10.1108/MD-07-2016-0460

[3] Wamba, S.F., Gunasekaran, A., Akter, S., Ren, S.J., Dubey, R., Childe, S.J. (2017) “Big data analytics and firm performance: Effects of dynamic capabilities”. Journal of Business Research, 70, pp. 356-365. DOI: https://doi.org/10.1016/j.jbusres.2016.08.009

[4] Ioannidis, J.P., Cripps, S., Tanner, M.A. (2020) “Forecasting for COVID-19 has failed”. International Journal of Forecasting, 38(2), pp. 423-438. DOI: https://doi.org/10.1016/j.ijforecast.2020.08.004

[5] Janssen, M., Van Der Voort, H., Wahyudi, A. (2017) “Factors influencing big data decision-making quality”. Journal of Business Research, 70, pp. 338-345. DOI: https://doi.org/10.1016/j.jbusres.2016.08.007

[6] N. Elgendy, A. Elragal T. Päivärinta (2021) “DECAS: a modern data-driven decision theory for big data and analytics”. Journal of Decision Systems, 31(4), pp. 337-373. DOI: https://doi.org/10.1080/12460125.2021.1894674

[7] Wills, T. (2019) “Sweden: Rogue algorithm stops welfare payments for up to 70,000 unemployed”. AlgorithmWatch, available online at: https://algorithmwatch.org/en/rogue-algorithm-in-sweden-stops-welfarepayments

[8] Spielkamp, M. (2019) “Automating Society: Taking Stock of Automated Decision-Making in the EU”. Algorithmwatch Report, BertelsmannStiftung Studies 2019.

[9] Mezias, J., Starbuck, W.H., (2009) “Decision making with inaccurate, unreliable data”. The Oxford handbook of organizational decision making, pp. 76-96, Oxford University Press.

[10] Amankwah-Amoah, J., Adomako, S. (2019) “Big data analytics and business failures in data-rich environments: An organizing framework”. Computers in Industry, 105, pp. 204-212. DOI: https://doi.org/10.1016/j.compind.2018.12.015

[11] Huang, X. Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E. Wu, M., Yi, X. (2020) “A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability”. Computer Science Review, 37. DOI: https://doi.org/10.1016/j.cosrev.2020.100270sdfsdfsdf

[12] Balayn, A., Lofi, C., Houben, G. J. (2021) “Managing bias and unfairness in data for decision support: a survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems”. The VLDB Journal, 30, pp. 739-768. DOI: https://doi.org/10.1007/s00778-021-00671-8

[13] Badii, C., Nesi, P., Paoli, I. (2018) "Predicting Available Parking Slots on Critical and Regular Services by Exploiting a Range of Open Data,". IEEE Access,6, pp. 44059-44071. DOI: https://doi.org/10.1109/ACCESS.2018.2864157

[14] Bellini, P., Cenni, D., Palesi, L. A. I., Nesi, P., Pantaleo, G. (2021) "A Deep Learning Approach for Short Term Prediction of Industrial Plant Working Status”. 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService), pp. 9-16, DOI: https://doi.org/10.1109/BigDataService52369.2021.00007

[15] United Nations (2018) “The 2030 Agenda and the Sustainable Development Goals: An opportunity for Latin America and the Caribbean” (LC/G.2681-P/Rev.3), Santiago.

[16] Attard-Frost, B., De los Ríos, A., Walters, D. R. (2022) “The ethics of AI business practices: a review of 47 AI ethics guidelines”. AI Ethics (2022). DOI: https://doi.org/10.1007/s43681-022-00156-6

[17] Vucenovic, A., Ali-Ozkan, O., Ekwempe, C., Eren, O. (2020) "A Case Study: Predicting Hospital Readmission Within 30 Days of Discharge". 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-4, DOI: https://doi.org/10.1109/CCECE47787.2020.9255721

[18] European Parliament, Directorate-General for Parliamentary Research Services, Lagioia, F. (2021) “The impact of the general data protection regulation on artificial intelligence”. Sartor, G.(editor), Publications Office.

[19] Badii, C., Bellini, P., Difino, A., Nesi, P., Pantaleo, G., Paolucci, M. (2019) “Microservices suite for smart city applications”. Sensors (Switzerland). 19(21). DOI: https://doi.org/10.3390/s19214798

[20] Badii, C., Bellini, P., Difino, A., Nesi, P. (2020) "Smart City IoT Platform Respecting GDPR Privacy and Security Aspects", IEEE Access, vol. 8, pp. 23601-23623. DOI: https://doi.org/10.1109/ACCESS.2020.2968741

[21] Collini, E., Ipsaro Palesi, A. L., Nesi, P., Pantaleo, G., Nocentini, N., Rosi, A. (2022) "Predicting and Understanding Landslide Events with Explainable AI". IEEE Access, 2022. DOI: https://doi.org/10.1109/ACCESS.2022.3158328

[22] Bellini, P., Cenni, D., Mitolo, N., Nesi, P., Pantaleo, G., Soderi, M. (2021) "High Level Control of Chemical Plant by Industry 4.0 Solutions". Journal of Industrial Information Integration, Elsevier. DOI: https://doi.org/10.1016/j.jii.2021.100276

[23] Collini, E., Nesi, P., Pantaleo, G. (2021) "Deep Learning for Short-Term Prediction of Available Bikes on Bike-Sharing Stations". IEEE Access.

[24] C. Badii, P. Bellini, D. Cenni, S. Chiordi, N. Mitolo, P. Nesi, M. Paolucci, "Computing 15MinCityIndexes on the basis of Open Data and Services", Proc. of the 2021 International Conference on Computational Science and Its Applications. Published on LNCS Springer.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Graphical Animations of an Autonomous Vehicle Merging Protocol*

Dang Duy Bui[a], Minxuan Liu[a], Duong Dinh Tran[a] and Kazuhiro Ogata[a,*]

[a]School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan

ARTICLE INFO

Article History:
Submitted 3.21.2023
Revised 3.30.2023
Accepted 4.3.2023

Keywords:
state machines
graphical animations
Maude
autonomous vehicle merging protocol
state picture template
Gestalt principles
lockout freedom property
counterexample visualization

ABSTRACT

State machine graphical animation (SMGA) is a tool that takes a state picture template and a state sequence of a state machine as inputs and generates a graphical animation of the state machine by replacing each state in the sequence with the corresponding state picture made from the state picture template as an output. SMGA helps humans to discover characteristics of state machines. r-SMGA is an integration of SMGA and Maude, a formal specification language/tool. Maude is equipped with various functionalities, such as a reachability analyzer (the search command) and an LTL model checker. r-SMGA makes it possible to use such Maude functionalities inside r-SMGA. We suppose that we understand a system/protocol so that we can write a formal specification of the system/protocol. Thus, we know some characteristics of the system/protocol, such as the values that characterize each state of the state machine. Characteristics that can be observed through the formal specification of a system/protocol are called shallow characteristics. Even shallow characteristics positively affect the quality of state picture templates. In this paper, we use an autonomous vehicle merging protocol as an example to demonstrate the claim. We also rely on some Gestalt principles to design state picture templates. Based on our design and Gestalt principle, we describe how to discover deeper characteristics of the state machine that formalizes the protocol with r-SMGA and how to filter out false characteristics with the search command available in r-SMGA if characteristics are likely invariant properties. In addition to invariant properties, there are some other important classes of properties with respect to state machines, such as leads-to properties. We change the behavior of each vehicle (such change does not affect the essence of the protocol) and find that the protocol does not enjoy a leads-to property with the model checker available inside r-SMGA, finding a counterexample. The loop part of the counterexample is graphically animated, which makes us comprehend reasons why the protocol does not enjoy the property and come up with a revised version that enjoys the property.

1. Introduction

State machine graphical animation (SMGA) [20] is a tool to visualize a system/protocol formalized as a state machine. The input of SMGA is a state picture template (designed by humans) and a state sequence (generated from a formal specification of a system/protocol by Maude, a formal specification language/tool [17]). The output is a graphical animation of the state machine by replacing each state in the sequence with the corresponding state picture made from the state picture template. SMGA can aid humans in discovering characteristics of systems/protocols [2, 7, 9, 18] through observing their graphical animations. r-SMGA [10] is an integration of SMGA and Maude. Maude is equipped with various functionalities, such as a reachability analyzer (the search command) and an LTL model checker, were LTL stands for linear temporal logic. r-SMGA makes it possible to use such Maude functionalities inside r-SMGA, and to automatically generate a state sequence from a Maude specification. The present paper reports on a case study in which r-SMGA is mainly used. Because designing state picture templates is the key task in r-SMGA [9], it is worth investigating this task. In this paper, we describe how to design state picture templates for r-SMGA by using a concrete non-trivial example, a revised version [16] of the autonomous vehicle merge protocol proposed by Aoki and Rajkumar [1]. The original protocol proposed by Aoki and Rajkumar is called

*The present paper is an extended and revised version of the paper [6] presented at DMSVIVA 2022

*Corresponding author
bddang@jaist.ac.jp (D.D. Bui); liuminxuan@jaist.ac.jp (M. Liu); duongtd@jaist.ac.jp (D.D. Tran); ogata@jaist.ac.jp (K. Ogata)
ORCID(s): 0000-0002-2700-1762 (D.D. Bui); 0000-0001-7092-2084 (D.D. Tran); 0000-0002-4441-3259 (K. Ogata)

DOI reference number: 10-18293/JVLC2023-N1-027

the AR protocol, while the revised version is called the r-AR protocol. The AR protocol depends on realtime information, while the revised version does not. The reason why Liu, et al. [16] made the revised version non-realtime was because they would like to focus on more basic mechanisms that have nothing to do with realtime and to this end, it would be reasonable to remove any realtime information on which the AR protocol relies. The goal of the AR protocol is to control autonomous vehicles to avoid crashing each other at a merge point where two lanes (a through lane and a non-through one) are merged, and so does the r-AR protocol.

To handle the goal of the r-AR protocol, the authors [16] use statuses for each vehicle. Let us briefly describe how each vehicle changes its status in the protocol. Each vehicle on each of the through and non-through lanes is initially in the *running* status, meaning that it is enough far from the merge point and it may go over the vehicles running in front of it. When it gets enough near the merge point, its status changes to the *approaching* one. When a vehicle is in the approaching status, we suppose that it never goes over the vehicles running in front of it. The vehicle on each lane just in front of the merge point will pass through the merge point or stop just before the merge point. In other words, if the former occurs, the vehicle's status changes to *crossing*; otherwise, it changes to *stopped*. Finally, when a vehicle passed the merge point, its status changes to *crossed* from the crossing status.

In the paper, we describe graphical animations of the r-AR protocol, where the main idea is to visualize each vehicle's status with its lane (i.e., through or non-through) based on Gestalt principles. We design state picture templates of the r-AR protocol based on these ideas and shallow characteristics of the protocol (obtained via its specification). By observing graphical animations, we conjecture some deep characteristics of the r-AR protocol to show that Gestalt principles is one factor affecting the design of state picture template.

In the formal verification of the r-AR protocol, we suppose that a small number of vehicles participate in the r-AR protocol, and each vehicle on each of the through and non-through lanes passes through the merge point once; and only consider invariant properties as desired properties. There are some more non-invariant desired properties of the r-AR protocol. One such possible property is that vehicles approaching the merge point or stopping just in front of the merge point on each lane will eventually pass through the merge point. The property belongs to a class of liveness properties, precisely leads-to properties, and is called the lockout freedom property in this paper. We use the Maude LTL model checker (integrated into r-SMGA) such that the r-AR protocol enjoys the lockout freedom property; and do not find any counterexamples. The result is not surprising because a small number of vehicles participate in the r-AR protocol and each vehicle passes through the merge point once. Thus, we modify the formal specification of the r-AR protocol such that each vehicle repeatedly tries to pass through the merge point. In other words, when each vehicle running on each lane has passed through the merge point, it goes back to the running status and repeatedly runs following the protocol's behavior. When we model check that such version enjoys the lockout freedom property, we find a counterexample whose form is a finite sequence of states plus a finite loop of states. We newly design the state picture template for the modified formal specification of the r-AR protocol. We also revise r-SMGA so that it can handle the finite loop of states in a counterexample. By observing the graphical animation of the finite loop of states, we notice reasons why the modified version does not enjoy the lockout freedom property. We then revise the r-AR protocol and model check that the revised version of the r-AR protocol enjoys the lockout freedom property, finding no counterexamples.

The present paper is an extended and improved version of the paper [6] accepted by DMSVIVA 2022. New contributions of the present paper that are not described in the DMSVIVA 2022 paper [6] are as follows:

- The formal specification of the r-AR protocol is modified such that each vehicle repeatedly tries to pass through the merge point;
- The state picture template is newly designed for the modified specification and new tips for designing state picture templates are discovered;
- A counterexample is found for the lockout freedom property for the r-AR protocol under the assumption that each vehicle tries to pass through the merge point repeatedly;
- r-SMGA is revised so that it can deal with the finite loop of states in a counterexample;
- By observing the graphical animation of the finite loop of states, we find reasons why the lockout freedom property is broken for the r-AR protocol under the assumption; and revise the r-AR protocol so that the lockout freedom property is satisfied under the assumption.

The rest of the paper is structured as follows. Section 2 mentions state machines, Maude, r-SMGA, and Gestalt principles as preliminaries. Section 3 introduces the r-AR protocol and its specification in Maude. In Section 4, we describe in detail how to design the state picture template of the r-AR protocol. By observing graphical animations generated based on the state picture template, we then guess some characteristics (likely invariant properties) of the r-AR protocol; and confirm them with the Maude search command integrated into r-SMGA in Section 5. Section 6 describes how we modify the formal specification of the r-AR protocol, model check the lockout freedom property of the modified protocol, show graphical animations of the counterexample, and explain reasons why the modified protocol does not enjoy such property. In Section 7, we revise the modified protocol so that it can enjoy the lock out freedom property. Section 8 mentions some related work. Finally, we conclude the present paper in Section 9.

2. Preliminaries

We briefly describe state machines, Maude, r-SMGA, and some Gestalt principles for readers to better follow the content of the paper.

2.1. State Machines and Maude

A state machine M is defined as $\langle S, I, T \rangle$, where S is a set of states, $I \subseteq S$ is the set of initial states, and $T \subseteq S \times S$ is a binary relation over states. We call $(s, s') \in T$ a state transition, where $s, s' \in S$. The set of reachable states with respect to M is inductively defined as follows: (1) $I \subseteq R$ and (2) if $(s, s') \in T$ and $s \in R$, then $s' \in R$. A state predicate p is an invariant property with respect to M if and only if $p(s)$ holds for all $s \in R$. A finite sequence of states $s_0, \ldots, s_i, s_{i+1}, \ldots, s_n$ is called a finite computation of M if $s_0 \in I$ and $(s_i, s_{i+1}) \in T$ for each $i = 0, \ldots, n-1$.

There are many possible ways to express a state $s \in S$. In the paper, we use a braced associative-commutative collection of name-value pairs. Associative-commutative collections are called soups, and name-value pairs are called observable components. Then, a state is expressed as a braced soup of observable components and the juxtaposition operator is used as the constructor of soups. Suppose $oc1, oc2, oc3$ are observable components, and then $oc1\ oc2\ oc3$ is the soup of those three observable components. A state can be expressed as $\{oc1\ oc2\ oc3\}$. There are also many possible ways to specify state transitions. One possible way is to use Maude [17], a programming/specification language based on rewriting logic, to specify them as rewrite rules. A conditional rewrite rule (or just a rule) is in the form as follows:

`crl [`lb`] :` l `=>` r `if ... /\` c_i `/\ ...`

where lb is the label given to the rule and c_i is a part of the condition, which may be an equation $lc_i = rc_i$. The negation of $lc_i = rc_i$ could be written as $lc_i =/= rc_i$

Maude provides the `search` command for reachability analysis that allows finding a state reachable from one state *init* such that the state matches the pattern p and satisfies the condition c. The command can be expressed as follows:

`search [`n`,`m`] in` MOD `:` $init$ `=>*` p `such that` c `.`

where n and m are a number of solutions and a bounding depth of a state space of a state machine under analysis, respectively; n is often 1 while m is often omitted meaning that the depth under traversal is not fixed; MOD is the name of the Maude module specifying the state machine; $init$, p, and c are the starting state, the pattern, and the condition, respectively. $init$ typically is an initial state of a state machine under verification. The search command can be used as an invariant model checker. The patten p and the condition c are used to express the negation of an invariant property or a state predicate under verification.

Maude is also equipped with an LTL model checker, where LTL stands for linear temporal logic. We suppose that readers are familiar with LTL and Kripke structures [11]. We need to use the LTL model checker so as to formally verify that a system/protocol formalized as a Kripke structure, an extension of state machines, enjoys a liveness property that can be expressed as an LTL formula. Note that invariant properties can be expressed as LTL formulas. When we would like to formally verify that such a system/protocol, where *init* is the only initial state, enjoys a liveness property expressed as an LTL φ, the following command is used:

`reduce modelCheck(`$init$`,` φ`) .`

Maude returns true if the system/protocol satisfies φ. Otherwise, a counterexample is returned, which has the form of a finite sequence of states and a finite loop of states. When a system/protocol has multiple initial states, it suffices to conduct the model checking experiment for each initial state so that we can model check that the system/protocol enjoys the property.

2.2. Revised State Machine Graphical Animation (r-SMGA)

Bui, et al. [10] have integrated SMGA and Maude, extending/revising SMGA. The extended/revised version of SMGA is called r-SMGA. Some functionalities of Maude, such as the search command and the LTL model checker, can be used inside r-SMGA. For r-SMGA, inputs are a formal specification of a system/protocol in Maude and a state picture template. r-SMGA allows human users to flexibly generate finite computations and store them as a list. For example, human users can give the depth of each finite computation being generated and generate the graphical animation of one of the finite computations generated before. r-SMGA also makes it possible to extract states from multiple finite computations generated so far by using a given associative-commutative (AC) pattern. Furthermore, r-SMGA allows human users to use some interactive features when observing graphical animations, such as focusing on some observable components that users are interested in.

r-SMGA uses DrawSVG [14] as SMGA does. Thus, r-SMGA as well as SMGA requires human users to design state picture templates. We suppose that a formal specification of a protocol/system under consideration has been written by a human user. Thus, a braced soup of observable component, how to express/formalize each state, can be used to design a state picture template. Our approach to designing a state picture template or visualizing a state is as follows. As mentioned, a state is formalized as a braced soup of observable components, such as $\{oc1\ oc2\ oc3\}$. Thus, it suffices to visualize each observable component (such as $oc1, oc2, oc3$) to visualize a state. Then, we design a visualization template for each observable component to design a state picture template. Multiple instances of an observable component may be used, and even if that is the case, it suffices to design a visualization template for each observable component but not for each instance of it, where an observable component corresponds to a type and its instances correspond to values of the type. Fixing values used in an observable component, an instance of the observable component is made, and then a visualization template for an observable component becomes a concrete visualization object. Basically, there are two possible ways to visualize observable components: (1) textual

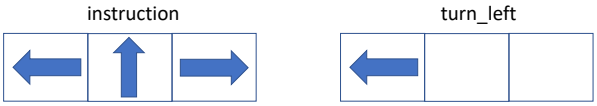


Figure 1: An example of an observable component *traffic-sign*, where the left-hand side is the state picture template and the right-hand side is one concrete state picture

Figure 2: An example before applying gestalt principles

display and (2) visual display. Note that an instance of an observable component can be displayed in both (1) and (2). Option (1) displays values of an observable component instance as texts, while option (2) displays them as visual objects where such visual objects (such as circles and arrows designed by humans) correspond to the values.

For example, let us consider an observable component `(traffic-sign: instruction)`, where `traffic-sign` is the name and *instruction* is the value that can be one of `turn_left`, `go_straight`, and `turn_right`. Therefore, there are the three instances: `(traffic-sign: turn_left)`, `(traffic-sign: go_straight)` and `(traffic-sign: turn_right)`. The observable component simulates a traffic sign that orders a vehicle to turn left, go straight or turn right. The left-hand side of Figure 1 shows a possible state picture template of the observable component, while the right-hand side shows the state picture of the instance `(traffic-sign: turn_left)`. For the state picture, the observable component is displayed in both (1) and (2) as seen.

2.3. Gestalt Principles

Gestalt principles [24, 25] are a collection of principles, such as the common region and the similarity principles, related to visual perception of humans about grouping. Let us use some concrete examples to describe the two principles in Gestalt principles. Taking a look at Figure 2, we can recognize that there are one group including six hearts. Based on the common region principle (grouping elements that are in the same closed region), Figure 3 can help us to recognize that there are three groups where each group contains two hearts. Similarly, based on the similarity principle (humans tend to build a relationship between similar elements via basic elements, such as color and size), Figure 4 and Figure 5 can also help us to recognize that there are three groups where each group contains two hearts, where Figure 4 uses color while Figure 5 uses size.

3. The r-AR Protocol

In this section, we first describe the r-AR protocol and its specification in Maude. We then introduce the observable components that are used to visualize the protocol.

Figure 3: An example after applying the common region principle

Figure 4: An example after applying the similarity principle (using color)

Figure 5: An example after applying the similarity principle (using size)

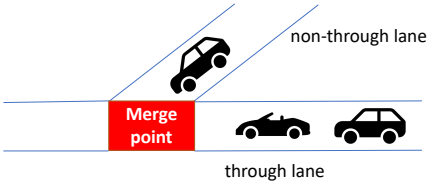


Figure 6: A merge point

3.1. Protocol Descriptions

S. Aoki and K. Rajkumar [1] have proposed an autonomous vehicle merging protocol (called the AR protocol) with two lanes called through and non-through lanes as depicted in Figure 6. The horizontal line refers to the through lane, the diagonal line refers to the non-through lane, and the intersection of those two lanes is called the merge point. Vehicles on both lanes are supposed to run toward the merge point and in one direction only. At the merge point, vehicles are controlled so that they never collide with each other. In other words, the protocol guarantees at most one vehicle located at the merge point.

In the original protocol, there are two versions corresponding to two traffic environments: (1) only autonomous vehicles on the traffic (homogeneous traffic), and (2) autonomous vehicles and human-driven vehicles on the traffic (heterogeneous traffic). Liu, et al. [16] have revised the first version such that the revised version (called the r-AR protocol) does not rely on any real-time information, such as speed of vehicles running on both lanes. There are two modes in the r-AR protocol: prioritized and fair. In the prioritized mode, vehicles on the non-through lane (or non-through lane vehicles) cannot enter the merge point if some vehicles on the through lane (or through lane vehicles) are approaching the merge point. Basically, there are three situations in the prioritized

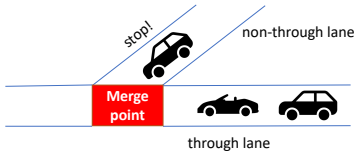


Figure 7: A case that a non-through lane vehicle stops before the merge point because of having some through lane vehicles approaching the merge point

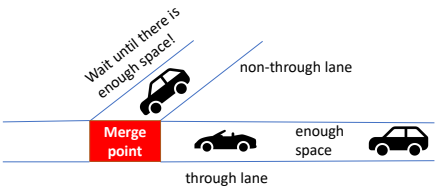


Figure 8: A case where there is enough space on the through lane for a non-through lane vehicle to enter the merge point

mode:

1. If some through lane vehicles are approaching the merge point and there is not enough space between any of two adjacent vehicles, then non-through lane vehicles must stop before the merge point until all through lane vehicles have passed through the merge point. Figure 7 is an example of this case.
2. If no through lane vehicle is approaching the merge point, non-through lane vehicles can enter the merge point.
3. If there is enough space between two adjacent through lane vehicles, then non-through lane vehicles can use the space to enter the merge point. Figure 8 is an example of this case.

When the traffic of the through lane becomes congested, the prioritized mode changes to the fair mode. In the fair mode, through and non-through lane vehicles can enter the merge point alternately. If the traffic of the through lane becomes less congested, the mode changes back to the prioritized mode.

In the r-AR protocol, each vehicle is assigned one of the following five values as its status:

- `running`: when a vehicle is far away from the merge point, its status is `running`.
- `approaching`: when a vehicle gets close to the merge point, its status changes from `running` to `approaching`.
- `stopped`: when a vehicle meets some conditions, it stops before the merge point and its status changes from `approaching` to `stopped`.
- `crossing`: if a vehicle has just entered the merge point, its status changes from `approaching` or `stopped` to `crossing`.
- `crossed`: when a vehicle has passed the merge point, its status changes from `crossing` to `crossed`.

Each vehicle whose status is `running` or `crossed` on each lane may go over those running on the same lane in front of it, while we suppose that each vehicle whose status is `approaching` never does so. Note that each vehicle whose status is `stopped` or `crossing` does not do so if the protocol works as intended. Therefore, we formalize the vehicles whose statuses are `approaching`, `stopped`, or `crossing` on each lane as a queue. In what follows, we will write that the through lane and the non-through lane are formalized as queues, which means that the vehicles on each lane whose statuses are `approaching`, `stopped` or `crossing` are maintained by putting them into a queue. The queue that consists of through lane vehicles is called the queue of the through lane or the through lane queue, while the queue that consists of non-through lane vehicles is called the queue of the non-through lane or the non-through lane queue. The through lane queue may consists of dummy vehicles. Congested traffic is defined as follows: the number of through lane vehicles whose statuses are `approaching` or `stopped` becomes greater than a specific number, making the mode fair. Otherwise, the mode is prioritized. In the fair mode, there is a turn for the through lane and the non-through lane. If the turn is through lane, then through lane vehicles have a higher priority to enter the merge point. If the turn is non-through lane, then non-through lane vehicles have a higher priority to enter the merge point. The turn is basically alternately changed between through lane and non-through lane in the fair mode. Therefore, vehicles on both lanes can enter the merge point alternately in the fair mode based on the turn.

The left-most flowchart in Figure 9 shows how statuses of through lane vehicles are changed. The changes between vehicle statuses, e.g., from `running` to `approaching`, are represented by arrows, possibly with some conditions that are not shown in the figure. The following describes in detail the conditions for changing the status of through lane vehicles. A vehicle status changes from `running` to `approaching` if the through lane vehicle gets close to the merge point. The status of a through lane vehicle approaching the merge point changes from `approaching` to `stopped`, which means that the vehicle must stop just before the merge point, if either:

- there is another vehicle at the merge point, or
- when the protocol is in the fair mode and currently the turn is non-through lane.

The status of a through lane vehicle approaching the merge point changes from `approaching` to `crossing`, which means that the vehicle enters the merge point, if either:

- when the protocol is in the prioritized mode and there is no vehicle at the merge point, or
- when the protocol is in the fair mode, the current turn is the through lane and there is no vehicle at the merge point.

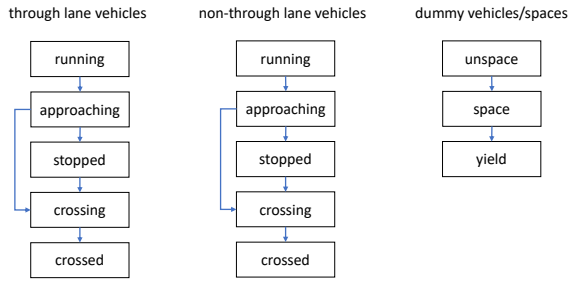


Figure 9: Transitions of status of through lane vehicles, non-through lane vehicles, and dummy vehicles (spaces).

The status of a through lane vehicle stopping the merge point changes from `stopped` to `crossing`, which means that the vehicle enters the merge point, if either:

- when the protocol is in the prioritized mode and there is no vehicle at the merge point, or
- when the protocol is in the fair mode, the current turn is through lane and there is no vehicle at the merge point, or
- when the protocol is in the fair mode, the current turn is non-through lane, there is no vehicle at the merge point, and there is no non-through lane vehicle in the non-through lane queue, which implies that there is no vehicle on the lane just before the merge point.

Vehicle status changes from `crossing` to `crossed` if the through lane vehicle whose status is `crossing` has passed the merge point.

Each space is assigned one of the three statuses: `unspace`, `space`, and `yield`, referring to the space that is not in the queue, is in the queue, and has just been out of the queue, respectively. Spaces on the through lane are treated as dummy vehicles running on the through lane, while it is unnecessary to explicitly deal with spaces on the non-through lane. The right-most flowchart in Figure 9 shows how the status of a space changes. As depicted in the figure, the status `unspace` cannot directly change to the status `yield`. For the change of the statuses of non-through lane vehicles (the middle flowchart in Figure 9) and dummy vehicles/spaces (the right-most flowchart in Figure 9), please refer to [16] in detail.

3.2. Formal Specification of the r-AR Protocol in Maude

Liu, et al. [16] have formally specified the r-AR protocol as a state machine in Maude. Two lanes that are close to the merge point are formalized as two queues of vehicles. The non-through lane queue consists of only real vehicles. Whereas, the non-through lane queue consists of real vehicles and spaces (also called dummy vehicles in this paper). The observable components used to formalize the r-AR protocol are as follows:

- `(lane[`l`]:` q`)` - l is one of `through` and `nonThrough`, corresponding to the through lane and the non-through lane, respectively. q is a queue of (possibly dummy) vehicle IDs. Initially, q is `empq` (denoting the empty queue) for both lanes.
- `(v[`id`]:` l, vs`)` - id is a vehicle ID (possibly a dummy vehicle ID). l and vs are the lane and the status information, respectively, of the vehicle ID. id is in the form of either `v(`i`)` when it is a real vehicle, or `dv(`i`)` when it is a dummy vehicle, where i is a natural number distinguishing the vehicle from each other. l is either `through` or `nonThrough`. For real vehicles, vs is `running`, `approaching`, `stopped`, `crossing`, or `crossed`; initially, vs is `running`. For dummy vehicles, vs is `unspace`, `space`, or `yield`; initially, vs is `unspace`.
- `(crossing?:` b`)` - it represents whether there exists a vehicle crossing the merge point. If so, b is `true`; otherwise, it is `false`. Initially, b is `false`.
- `(mode:` m`)` - it represents the mode in the r-AR protocol. m is one of `prioritized` and `fair` corresponding to the prioritized mode and the fair mode, respectively. Initially, m is `prioritized`.
- `(turn:` l`)` - it represents the turn when the system is in the fair mode. l is one of `through` and `nonThrough` corresponding to the turn of through lane and non-through lane, respectively. Initially, l is `through`.
- `(#uvcs:` n`)` - it represents the number of vehicles that have not yet passed the merge point. n is a natural number. Initially, n equals the number of vehicles participating in the protocol. It is reduced by one when a vehicle has just passed the merge point. When n is 0, all vehicles have crossed the merge point.
- `(gstat:` f`)` - it indicates that all vehicles have passed the merge point. When all vehicles have passed the merge point, f is `fin`; otherwise, it is `nFin`. Initially, f is `nFin`.

Suppose that there are two vehicles participating in the non-through lane, and four vehicles and two spaces participating in the through lane. If so, the initial state can be expressed as follows:

```
(gstat: nFin) (#ucvs: 6) (crossing?: false)
(mode: prioritized) (turn: through)
(lane[through]: empq) (lane[nonThrough]: empq)
(v[v(0)]: through,running) (v[v(1)]: through,running)
(v[v(2)]: through,running) (v[v(3)]: through,running)
(v[v(4)]: nonThrough,running) (v[dv(0)]: through,unspace)
(v[v(5)]: nonThrough,running) (v[dv(1)]: through,unspace)
```

As mentioned in the previous sub-section, there are some specific conditions on which a status of a vehicle (including a dummy one) changes to another. In the formal specification, rewrite rules in Maude are used to describe state transitions,

in which such conditions are embedded. We show here one of the rewrite rules with which the status of a through lane vehicle changes to `crossing` from `stopped`:

```
rl [enter-fairN-T] :
 {(v[v(I)]: through,stopped) (lane[through]: (v(I) ; TQ))
 (lane[nonThrough]: empq) (mode: fair)
 (turn: nonThrough) (crossing?: false) OCs}
 => {(v[v(I)]: through,crossing) (mode: fair)
 (lane[nonThrough]: empq) (turn: nonThrough)
 (lane[through]: (v(I) ; TQ)) (crossing?: true) OCs} .
```

where `I`, `TQ`, and `OCs` are Maude variables of natural numbers, queues, and soups of observable components, respectively. The rewrite rule says that if `mode` is `fair`, `turn` is `nonThrough`, `crossing?` is `false`, the non-through lane queue is empty, and the top vehicle on the through lane is `v(I)` whose status is `stopped`, then `v(I)` changes its status to `crossing`, and `crossing?` is updated to `true`. Meanwhile, in the fair mode, the non-through lane vehicle `v(I)` goes into the merge point when no through lane vehicle is the through lane. The other rewrite rules can be written likewise. The complete specification of the protocol is available at the URL[1].

4. Designing a State Picture Template of the r-AR Protocol

Designing state picture templates is the key task in r-SMGA and also a non-trivial task [9]. If a state picture template is too simple (it contains texts only [8]) or too complex (it contains many values for each observable component [9]), it takes so much effort to conjecture characteristics when observing graphical animations. This section first describes how to design the state picture template using the r-AR protocol as an example. We then show some factors affecting the design of the state picture template. The state picture template of the r-AR protocol is finally shown.

4.1. Idea

We suppose that we have formally specified a system/protocol as a state machine under visualization as described, namely that each state is formalized as a braced soup of observable components (precisely observable component instances) and state transitions are written in rewrite rules. The assumption implies that we comprehend the system/protocol such that we can make such a formal specification in Maude. We can design a state picture template based on observable components. Some previous work [7, 8, 9] have pointed out the usefulness of their state picture templates and also given some tips to make a good state picture template. In the present paper, we mainly use some of such tips for our design and summarize them as follows:

- Values of observable components should be visualized as much as possible.
- When an observable component has only two kinds of values, it should be visually/graphically represented as a light bulb.
- If a value of an observable component does not change, it should be expressed as a fixed label.

It is not necessary to visualize all observable components used. For example, we do not need to visualize the observable component `(crossing?: b)` (hereinafter referred to as `crossing?` and the same for other observable components). This is because we can easily know its value by checking if there is a vehicle at the merge point. We can also easily know the value of the observable component `#ucvs` by checking if there exist some vehicles on the two lanes. Hence, we do not need to visualize such observable component. Each of the observable components `turn`, `gstat`, and `mode` should be visualized as a light bulb as described. The lane information of each vehicle (either `through` or `nonThrough`) should be visualized as a fixed label. The remaining observable components needed to be visualized are the observable components `lane[through]`, the `lane[nonThrough]`, and `v[`vid`]`. In the rest of this section, we describe how to visualize them and show the final version of a state picture template of the r-AR protocol.

Furthermore, by observing graphical animations based on earlier drafts of the state picture template of the r-AR protocol, we comprehend that when the mode is prioritized, the `turn` observable component does not affect vehicles entering the merge point. Note that this shallow characteristic can be extracted from specification, however, there are many shallow characteristics that may be overlooked by human users. Therefore, we design two observable components `turn` and `mode` following this characteristic. The idea is that when the mode is prioritized, we do not display the observable component `turn`. Note, to design the state picture template in the present paper, we use some Gestalt principles [24, 25] (such as, the common region principle for elements in the queues of both lanes, and the similarity principle using colors for the status of vehicles, e.g., `stopped`). In the next sub-section, we describe designs of observable components and a state picture template of the r-AR protocol.

4.2. Designing a State Picture Template

Figure 10 shows fully our proposed state picture template. We suppose that there are four vehicles and two spaces participating in the through lane, and two vehicles participating in the non-through lane based on the `init` mentioned in Section 3. Figure 11 is our idea for visualizing the through and non-through lanes with vehicles. In the figure, the vertical rectangle and the horizontal rectangle represent the non-through and through lane, respectively; the red region represents the merge point; two arrows represent two meanings: (i) the directions and (ii) the turns of two lanes. The shapes of the arrows refer to (i) while the colors of the arrows refer to (ii), where the light-yellow color and the light-blue color indicate the turn of non-through and through lane vehicles, respectively; circles with numbers inside represent ve-

[1] https://github.com/rSMGA/AVMP/blob/main/avmp.maude

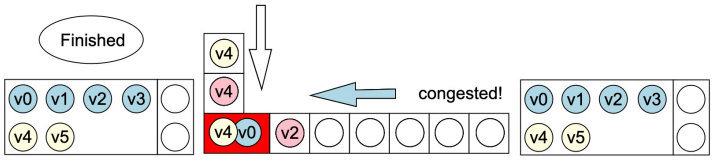


Figure 10: A state picture template for the r-AR protocol (1)

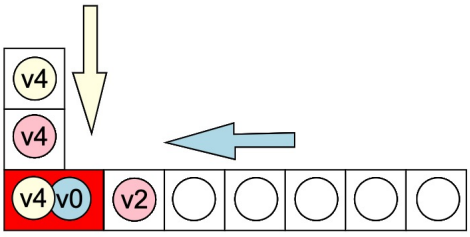


Figure 11: An idea of visualization of through and non-through lanes with vehicles

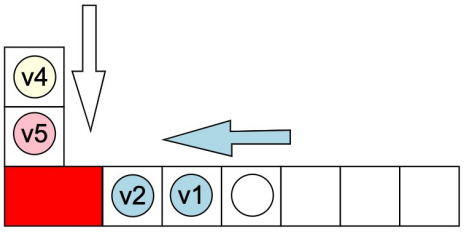


Figure 12: A concrete example of our proposed visualization for both lanes with vehicles

hicles with their IDs inside; white (or blank) circles represent spaces; blue and yellow circles represent through and non-through lane vehicles, respectively; pink circles represent vehicles whose status is `stopped`. Figure 12 shows two lanes when the value of the observable component `lane[through]` is `v(2);v(1);dv(1)` and the value of the observable component `lane[nonThrough]` is `v(5);v(4)`, where the semi-colon is used as the constructor of non-empty queues and the left-most is the top of each queue. In the figure, the status of `v(1)`, `v(2)`, `v(4)`, `v(5)`, and `dv(1)` are `approaching`, `approaching`, `approaching`, `stopped`, and `space`, respectively. Note that our design can help users to recognize a case such that two vehicles on both lanes collide at the merge point.

In Figure 10, as mentioned, both arrows indicate the turn of both lanes. In the prioritized mode, the light-blue arrow (nearing the through lane) and the white arrow (nearing the non-through lane) are displayed. In the fair mode, the text "congested!" is displayed and the arrows are displayed following the observable component `turn`, for example, if the value of `turn` is `nonThrough`, the light-yellow arrow (nearing the non-through lane) and the white arrow (nearing the through lane) are displayed. The rectangle in the left-most side of Figure 10 contains the vehicles and the spaces whose statues are `crossed` and `yield`, respectively. The rectangle in the right-most side of Figure 10 contains the vehicles and the spaces whose statues are `running` and `unspace`, respectively. The oval with the text "Finished" inside refers to the observable component `gStat`. When the value of `gStat` is `fin`, the oval is displayed; otherwise, it is not displayed.

Finally, our proposed design can be extended for more vehicles participating in the protocol. Users can design more squares when the number of squares can meet a case such that all vehicles are in the lane at the same time. For example, there are five vehicles and five spaces participating in the through lane, we need to prepare 10 positions (excluding the merge point) for a case that all of them are put into the queue. We prepare the state picture template shown in Figure 13 for a case in which there are five vehicles and five spaces in the through lane, and five vehicles in the non-through lane. Users can utilize it when the number of vehicles is up to five. Figure 14 shows a case that all through lane vehicles are in the queue. Note that we fix IDs of vehicles in the through and non-through lane from `0` to `4` and `5` to `9`, respectively. Therefore, users need to configure the initial state for such restriction.

5. Confirmation of Guessed Characteristics of the r-AR Protocol and Some Lessons Learned

In this section, we show the usefulness of our proposed design with factors (such as Gestalt principles) via conjecturing characteristics of the r-AR protocol. Finally, we confirm such characteristics by Maude search command integrated into r-SMGA.

5.1. Guessing Characteristics of the r-AR Protocol

Let us repeat that all examples are generated from the `init` as shown in Figure 10. We may recognize some characteristics of a protocol/system when formally specifying it. Graphical animations of the protocol/system make it possible to re-confirm such characteristics that are called shallow characteristics in the paper. It is worth doing so because formal specifications and state picture templates are supposed to be made by human beings who are subject to errors. Human errors may be detected by re-confirming shallow characteristics through observing graphical animations. For example, the following are two such shallow characteristics of the r-AR protocol that can be re-confirmed by ob-

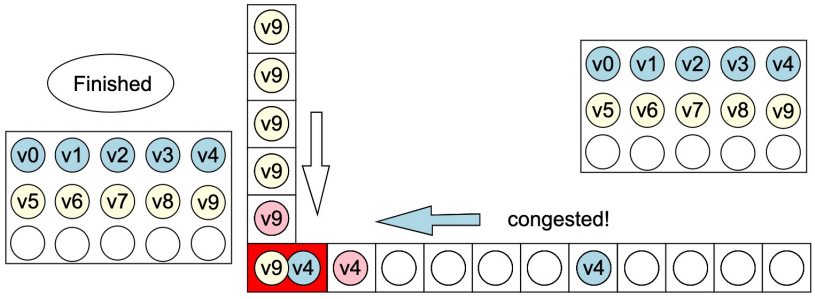


Figure 13: A state picture template for the r-AR protocol (2)

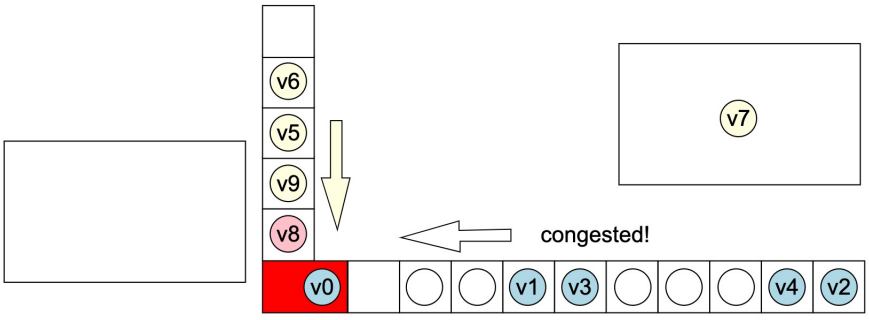


Figure 14: A state picture of when all through lane vehicles are in the queue

serving graphical animations, such as one of them shown in Figure 15:

- Characteristic 0.1: The turn is not concerned when the protocol is in the prioritized mode.
- Characteristic 0.2: There exists a case such that there is one vehicle whose status is `approaching` in the non-through lane, but this status changes to `stopped` even no vehicle is in the through lane.

Note that there are two sequences (two consecutive states for each) shown in Figure 15 obtained from two different sequences of states generated from the formal specification.

To conjecture some other characteristics, we use some tips [9]. They say that focusing on one or two observable components can help us to guess some relations of such observable components. We concentrate on two lanes and discover characteristics, some of which are as follows:

- Characteristic 1: There is at most one vehicle whose status is `stopped` in each lane.
- Characteristic 2: There are two vehicles whose statuses are `stopped` on both lanes, respectively, no vehicle is at the merge point.
- Characteristic 3: There are at most two vehicles whose statuses are `stopped` in the protocol.

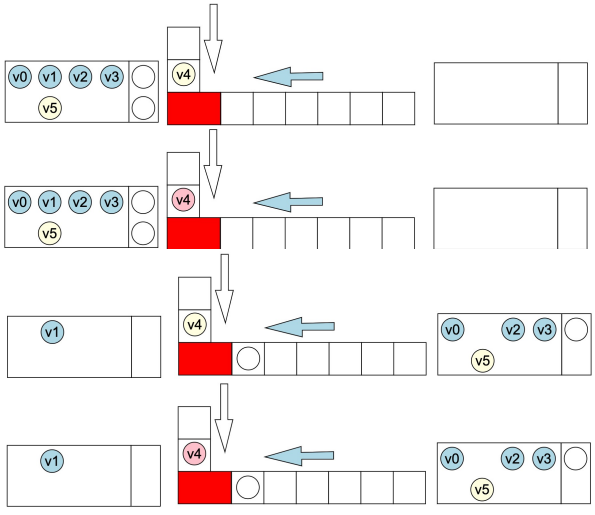


Figure 15: Some state pictures for Characteristic 0.2

To find the characteristics above, color is the main factor. Based on the colors designed based on the similarity principle, we can observe that there exists at most one light-pink color on each lane shown in Figure 16. Characteristic 3 can be conjectured based on two characteristics 1 and 2. The following characteristics are found by focusing on vehicles whose status is `stopped`.

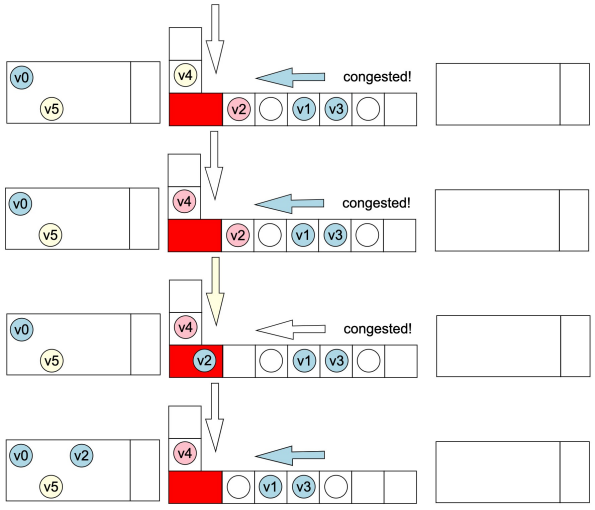


Figure 16: A piece of a finite computation

- Characteristic 4.1: If there is one vehicle whose status is `stopped` on the non-through lane and some vehicle is at the merge point, then this vehicle is on the through lane or the through lane queue is not empty.
- Characteristic 4.2: If there is one vehicle whose status is `stopped` on the through lane and some vehicle is at the merge point, then this vehicle is on the non-through lane or the non-through lane queue is not empty.

5.2. Confirmation of Guessed Characteristics

Characteristics guessed with r-SMGA may be wrong. The search command functionality available in r-SMGA makes it possible to check if there exists a counterexample for each characteristic. Note that we do not prove each characteristic and then a guessed characteristic for which no counterexample is found with the functionality may be wrong. Proof of characteristics for protocols/systems is out of the scope of the pater. For those who are interested in the topic, please refer to [5]. To sum up, when characteristics are confirmed (no counterexample is found) in the section, they are guaranteed for the case whose initial state is expressed as `init` as shown in Figure 10. The syntax of the functionality is the same as what is mentioned in Section 2. The following command is used to confirm Characteristic 1.

```
search [1] in AVMP : init =>*
{(v[X:Vid]: l1:Lane, stopped)
(v[Y:Vid]: l1:Lane, stopped) OCs:Soup{OComp}} .
```

where `AVMP` is the name of Maude module; `X:Vid` and `Y:Vid` are Maude variables of vehicle IDs; `l1:Lane` is a Maude variable of lanes; and `OCs:Soup{OComp}` is a Maude variable of observable component soups. The search command tries to find a reachable state from `init` in which there are two different vehicles whose statuses are `stopped` on one lane. If there is no such a reachable state from `init`, Characteristic 1 is an invariant property for the case whose initial state is `init`. It does not return any solution, meaning that no counterexample is found for the characteristic. Note that we use the Maude search command integrated into r-SMGA as shown in Figure 17 for Characteristic 1.

To confirm Characteristics 2 and 3, we use two commands as shown in Figure 18 and Figure 19, respectively. Each search command tries to find a reachable state that satisfies the corresponding pattern. They do not return any solution, meaning that no example is found for the two characteristics. Similarly, to confirm Characteristics 4.1 and 4.2, we use two commands as shown in Figure 20 and Figure 21, respectively. The two commands also do not return any solution, meaning that no counterexample is found for the two characteristics.

Note that we need to confirm all guessed characteristics because human users may overlook flawed cases. One flawed characteristic we have conjectured is as follows: when there is a non-through lane vehicle whose status is `approaching`, the next status of the vehicle is always stopped whenever its status changes. The characteristic cannot be expressed as an invariant property and then it is impossible to check the characteristic with the search command. We should use the Maude LTL model checker that is available in r-SMGA. When we use the Maude model checker in r-SMGA to confirm it, r-SMGA returns a counterexample. The counterexample says that the status of a non-through lane vehicle can change to `crossing` from `approaching` when the `mode` is `fair` and the `turn` is `nonThrough`.

6. A Flawed Case of the r-AR Protocol

Beside invariant properties of a state machine, there are liveness properties, such as leads-to properties. In this section, we will check whether the r-AR protocol satisfies a property called the lock-out freedom that can be expressed as a leads-to property. An informal description of the property says that when a vehicle approaches or stops just before the merge point, it will eventually go into the merge point.

If there are a few vehicles running on the two lanes and each vehicle tries to go through the merge point once, the r-AR protocol enjoys the lock-out freedom property. Therefore, we modify the behavior of each vehicle as follows: when the status of each vehicle is `crossed`, it will change to `running`, meaning that each vehicle tries to go through the merge point repeatedly. Note that we do not essentially change the r-AR protocol. To do it, we add the following rewrite rule:

```
rl [return] :
  {(v[v(I)]: L1,crossed) OCs} =>
  {(v[v(I)]: L1,running) OCs} .
```

where `return` is the name of the rule, and `I` and `L1` are Maude variables of vehicles IDs and lanes, respectively. The rule says that if the status of a vehicle `I` on a lane (through or non-through) is `crossed`, it changes to `running`. To model check the lock-out freedom property, we define some propositions as follows:

Figure 17: A command in r-SMGA to confirm Characteristic 1

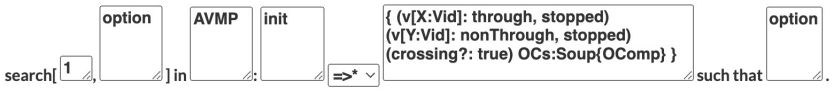


Figure 18: A command in r-SMGA to confirm Characteristic 2

```
eq {(v[v(I)]: L,approaching) OCs} |= approaching(I,L)
                                                = true .
eq {(v[v(I)]: L,stopped) OCs} |= stopped(I,L) = true .
eq {(v[v(I)]: L,crossed) OCs} |= crossed(I,L) = true .
eq {OCs} |= PROP = false [owise] .
```

where `_|=_` is a Maude operator for assigning the state (the first parameter) to a proposition (the second parameter) and returns a Boolean value. `approaching(_,_)`, `stopped(_,_)`, and `crossed(_,_)` are Maude operators that denote the propositions meaning that the status of a vehicle on a lane is `approaching`, `stopped`, and `crossed`, respectively. `PROP` is a Maude variable of propositions. The first equation says that the proposition `approaching(I,L)` is `true` if a status of the vehicle `I` on lane `L` is `approaching`; Propositions `stopped(I,L)` and `crossed(I,L)` are defined likewise. `owise` stands for otherwise. We define the lock-out freedom property as follows:

```
eq lockout-free(I,L) =
  (approaching(I,L) \/ stopped(I,L)) |-> crossed(I,L) .
```

where `_|->_` is a Maude operator to denote the leads-to temporal connective. The formula says that whenever the status of the vehicle `I` on the lane `L` is `approaching` or `stopped`, the status will eventually become `crossed`.

We use the Maude model checking functionality in r-SMGA to check the property for each lane and each vehicle. For any vehicles on the through lane, no counterexample is returned. It implies that the r-AR protocol satisfies the lock-out freedom property for the through lane. However, a counterexample is returned when model checking for the non-through lane, which says that the protocol does not satisfy the lock-out freedom property for the non-through lane. Figure 22 shows the state pictures representing the loop part of the counterexample, where two vehicles on the non-through lane never enter the merge point. The following is the order of the loop. The first state of the loop is shown at the top-most on the left column, the second state is shown at the top-most on the right column, the third state is shown at the second row of the left column, etc. The next state of the final state of the loop shown at the bottom-most on the right column is the first state. Note that we have deleted the two observable components `gstat` and `#uvcs` in the protocol and in the state picture template because each vehicle tries to go through the merge point repeatedly and then we do not need to maintain them. Taking a look at those state pictures via graphical animations, we can observe situations where two vehicles on the non-through lane never go through the merge point. r-SMGA allows human users to observe the loop of the state shown in Figure 22 as graphical animations. Careful observation of the graphical animations helps us to know that there are two cases in the states in the loop:

1. The mode is fair and a through lane vehicle is passing the merge point. Therefore, even though the turn is non-through, the non-through lane vehicle just before

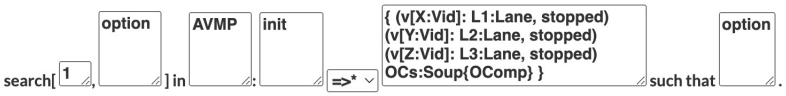


Figure 19: A command in r-SMGA to confirm Characteristic 3

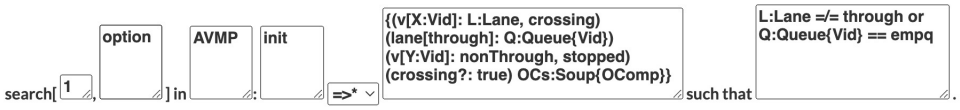


Figure 20: A command in r-SMGA to confirm Characteristic 4.1

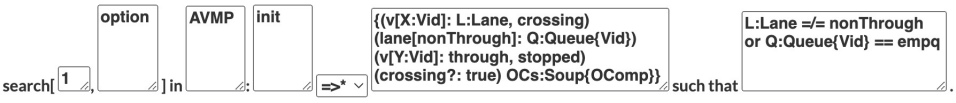


Figure 21: A command in r-SMGA to confirm Characteristic 4.2

the merge point is not allowed to cross the merge point.

2. The mode is prioritized. If this is the case, a vehicle is passing the merge point or there is a through lane vehicle just before the merge point. Hence, the non-though lane just before the merge point has no chance to cross the merge point.

In Figure 22, there are four states that belong to Case 1, where you can see "Congested!" appearing. For each of the four states, its next state is in the prioritized mode. The r-AR protocol cannot make the best use of the fair mode, letting a non-through lane vehicle to cross the merge point. For each state that belongs to Case 2, a through lane vehicle is crossing the merge point or there is a through lane vehicle just in front of the merge point. Therefore, non-through vehicles are never allowed to cross the merge point. In the next section, we revise the r-AR protocol so that the it can enjoy the lockout freedom property.

7. A Revised Version of the r-AR Protocol

As written, the r-AR protocol does not make the best use of the fair mode, letting a non-through lane to cross the merge point. In the r-AR protocol, where the number of the through lane vehicles whose status is `approaching` is greater than a fixed number, the mode becomes fair, but if the number becomes less than the fixed number, the mode becomes back prioritized. Between two modes, nothing guarantees that a non-through lane vehicle can pass the merge point. Therefore, we modify how to change the prioritized mode to the fair mode and vice versa.

To handle such situation, we use how many through lane vehicles have entered the merge point instead. Let N be such number. When N becomes greater than a fixed number, the mode is changed to fair from prioritized, and non-through lane vehicles are given a higher priority than though lane vehicles in the fair mode. If the through lane queue is empty, the non-through lane just before the merge point is allowed to enter the merge point. So, let us suppose that the through lane queue is not empty and the mode is prioritized. If that is the case, though lane vehicles cross the merge point, increasing N. When N becomes larger than a fixed number, the mode changes to fair from prioritized. If the non-through lane queue is not empty, the top vehicles of the queue is given the highest priority to enter the merge point and then the non-through lane vehicle is allowed to enter the marge point. We add one observable component `(#tlvp N)` representing the number of through lane vehicles that have passed the merge point. The rewrite rule used when a through lane vehicle enters the merge point in the prioritized mode is modified as follows:

```
rl [enter-prio-T] :
  {(v[v(I)]: through,VS) (lane[through]: (v(I) ; TQ))
  (crossing?: false) (mode: prioritized) (#tlvp: N) OCs}
   => {(v[v(I)]: through,crossing)
   (lane[through]: (v(I) ; TQ)) (crossing?: true)
   (mode: (if N < 2 then prioritized else fair fi))
   (#tlvp: ( if N < 2 then s(N) else N fi)) OCs} .
```

where `TQ` is a Maude variable of queues, `N` is a Maude variable of natural numbers, and `s(_)` is the successor function of natural numbers. We use `2` as the fixed number and can use a different positive number.

When a non-through lane vehicle has entered the merge point in the fair mode, we change the mode to prioritized from fair and make `n` used in `(#tlvp n)` 0. Then, the rewrite rule used when a non-through lane vehicle enters the merge point in the fair mode is revised as follows:

```
rl [enter-fair-N] :
  {(v[v(I)]: nonThrough,stopped)
  (lane[nonThrough]: (v(I) ; NQ))
  (mode: fair) (#tlvp: N) (crossing?: false) OCs}
   => {(v[v(I)]: nonThrough,crossing)
  (lane[nonThrough]: (v(I) ; NQ)) (mode: prioritized)
  (#tlvp: 0) (crossing?: true) OCs} .
```

where `NQ` is a Maude variable of queues.

We check if the revised version of the r-AR protocol satisfies that lockout freedom property with the model checking functionality in r-SMGA, and then no counterexample is found. We confirm that this revised version satisfies the lockout freedom property.

Note that we do not use the observable component `turn` because we use the observable component `mode` instead. The formal specification of the revised version of the r-AR protocol is available at the URL[2].

8. Related Work

Bui, et al. [7] have used SMGA to graphically animate the intersection traffic control distributed mutual exclusion protocol or the LJPL protocol [15]. In the protocol, there are eight lanes in which each two of them can be conflicted or concurrent. Vehicles in the concurrent lanes are allowed to enter the intersection at the same time while vehicles in the conflicted lanes are prohibited. Moreover, vehicles in the

[2] https://github.com/rSMGA/AVMP/blob/main/avmp-rev.maude

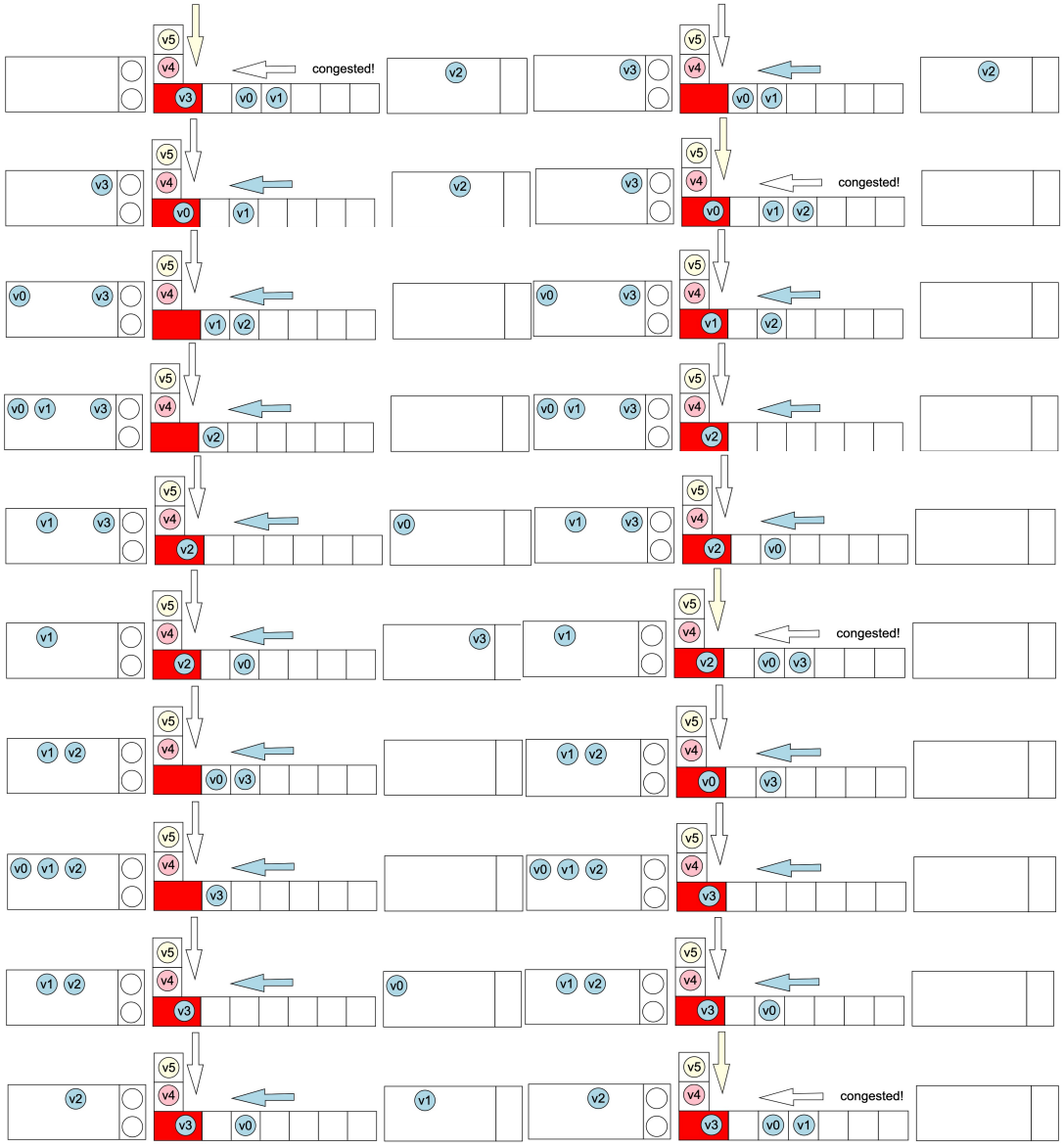


Figure 22: A loop generated by the model checking functionality in r-SMGA.

same lane can also enter the intersection at the same time. To guarantee the mutual exclusion property (e.g., no conflicted lane vehicles in the intersection), the protocol uses statuses for each vehicle (e.g., approaching and crossing) and controls them by their proposed algorithm. Based on some assumptions of the protocol, such as lanes of vehicles and vehicles' statuses, Bui, et al. [7] have revised SMGA and designed a state picture template so that the tool can visualize elements in the lanes (vehicles' IDs) followed by other elements (vehicles's statuses). By observing graphical animations based on their state picture template, the authors conjecture some non-trivial characteristics of the protocol and the characteristics have been confirmed with Maude. The r-AR protocol and the LJPL protocol share some ideas to design, such as using lanes of vehicles and statuses of vehicles, but, some assumptions of two protocols are different, and then we cannot apply all ideas to visualize the LJPL protocol for visualizing the r-AR protocol.

Frank, et al. [13] have proposed a method to visualize state transition systems of protocols/systems. They aim to let users observe global properties of protocols/systems by visualizing whole state spaces. They use cone tree [21] to form state transition structures in three dimensions so that users can observe the visualization in three dimensions. The main algorithm of the method focuses on the symmetry property and aims to let users identify the symmetrical and similar sub-structures in the tree. To do that, they first rank all nodes to make the systems become hierarchical system structures [23]. Then, they cluster such nodes following some local properties to reduce the visual complexity of the tree. Based on the clusters, they aim to visualize the whole state spaces as a backbone tree, where clusters are visualized as circles whose

sizes are decided by their volumes. Then, users can observe and interact with the tree by focusing and zooming into some clusters to be able to analyze paths inside. The method is extended to deal with a large state space [12]. The results of both versions allow users to observe state spaces of protocols visualized as a backbone tree with the cone tree concept for each node. Then users can find some global properties of the protocols, such as obtaining some clusters that do not return to initial nodes after starting some executions. This method and r-SMGA share the idea to help users find properties or characteristics of protocols. r-SMGA graphically animates state sequences (paths) while this method visualizes whole state spaces. The idea of the method motivates us to extend r-SMGA so that r-SMGA can give users an overview of state spaces. Then users can select some paths and graphically animate them by our approach. One piece of our future work is to extend r-SMGA based on the mentioned ideas.

r-SMGA can be considered a way to comprehend counterexamples via graphical animations where the counterexamples are returned by Maude. Nguyen, et al. [19] have shown that observing graphical animations of a shorter counterexample can make humans easier to comprehend. The idea is to use meta level in Maude to generate a shorter counterexample based on the search command in Maude. First, they use a model checker equipped with Maude to generate a counterexample when a system does not satisfy some property. Then, they use meta-search in Maude that uses a breadth-first search way (this search can make the counterexample shorter) to find a shorter state sequence leading the loop part of the counterexample. For the loop, it can work likewise. Finally, the shorter counterexample is graphically animated with SMGA.. In the present paper, the size of the counterexample is large (over 450 states) and then the approach to making a counterexample shorter would be useful for r-SMGA as well. It is one piece of our future work to utilize the approach in r-SMGA.

9. Conclusion

We have graphically animated the r-AR protocol with r-SMGA. Designing a state picture template is a non-trivial task in r-SMGA. To design the state picture template of the r-AR protocol, we have used the tips [9] and Gestalt principles [24, 25]. We also have shown some factors that affect the state picture template, such as colors in the similarity principles of Gestalt principles. Observing graphical animations helps us to reconfirm some shallow characteristics of the r-AR protocol that have been noticed when writing the formal specification and guess some deep characteristics of the protocol. Those characteristics have been confirmed with the Maude search command in r-SMGA. We have checked that the r-AR protocol does not enjoy the lockout freedom property using the Maude model checking function available in r-SMGA. By observing graphical animations of the counterexample, we can comprehend reasons why the protocol does not satisfy the property and revise the r-AR protocol so that the revised version enjoys that property. There are two main pieces of our future directions: one is to implement some factors that help humans to better comprehend graphical animations of a counterexample, such as size and causality [3]; another direction is to conduct more advanced case studies, such as quantum teleportation protocol [4] and Shor algorithm [22], to demonstrate the usefulness of our approach [5].

References

[1] Aoki, S., Rajkumar, R.R., 2017. A merging protocol for self-driving vehicles, in: ICCPS, p. 219–228. doi:10.1145/3055004.3055028.

[2] Aung, M.T., Nguyen, T.T.T., Ogata, K., 2018. Guessing, model checking and theorem proving of state machine properties – a case study on Qlock. IJSECS 4, 1–18. doi:10.15282/ijsecs.4.2.2018.1.0045.

[3] Beer, I., Ben-David, S., Chockler, H., Orni, A., Trefler, R., 2012. Explaining Counterexamples Using Causality. Formal Methods in System Design 40, 20–40. doi:10.1007/s10703-011-0132-2.

[4] Bennett, C.H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., Wootters, W.K., 1993. Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels. Phys. Rev. Lett. 70, 1895–1899. doi:10.1103/PhysRevLett.70.1895.

[5] Bui, D.D., 2022. State Machine Visualization Based on Gestalt Principles and Its Applications. Ph.D. thesis. Japan Advanced Institute of Science and Technology. URL: http://hdl.handle.net/10119/18189.

[6] Bui, D.D., Liu, M., Ogata, K., 2022a. Graphical Animations of an Autonomous Vehicle Merging Protocol, in: DMSVIVA 2022, KSI Research Inc.. pp. 16–22. doi:10.18293/DMSVIVA22-009.

[7] Bui, D.D., Myint, W.H.H., Tran, D.D., Ogata, K., 2022b. Graphical animations of the Lim-Jeong-Park-Lee autonomous vehicle intersection control protocol. JVLC 2022, 1–15. doi:10.18293/JVLC2022-N1-004.

[8] Bui, D.D., Ogata, K., 2019. Graphical animations of the Suzuki-Kasami distributed mutual exclusion protocol. JVLC 2019, 105–115. doi:10.18293/JVLC2019-N2-012.

[9] Bui, D.D., Ogata, K., 2022. Better state pictures facilitating state machine characteristic conjecture. Multimedia Tools and Applications 81, 237–272. doi:10.1007/s11042-021-10992-z.

[10] Bui, D.D., Tran, D.D., Ogata, K., Riesco, A., 2022c. Integration of SMGA and Maude to Facilitate Characteristic Conjecture, in: DMSVIVA 2022, KSI Research Inc.. pp. 45–54. doi:10.18293/DMSVIVA22-006.

[11] Clarke, E.M., Grumberg, O., Kroening, D., Peled, D.A., Veith, H., 2018. Model checking, 2nd Edition. MIT Press. URL: https://mitpress.mit.edu/books/model-checking-second-edition.

[12] Groote, J., Ham, F., 2006. Interactive visualization of large state spaces. STTT 8, 77–91. doi:10.1007/s10009-005-0198-5.

[13] van Ham, F., van de Wetering, H., van Wijk, J.J., 2002. Interactive Visualization of State Transition Systems. IEEE Transaction on Visual and Computer Graphics 8, 319–329. doi:10.1109/TVCG.2002.1044518.

[14] Liard, J., . Draw-svg the free online drawing tools. https://www.drawsvg.org/. Accessed: 2022-04-25.

[15] Lim, J., Jeong, Y., Park, D., Lee, H., 2018. An efficient distributed mutual exclusion algorithm for intersection traffic control. J. Supercomput. 74, 1090–1107. doi:10.1007/s11227-016-1799-3.

[16] Liu, M., Bui, D.D., Tran, D.D., Ogata, K., 2021. Formal specification and model checking of an autonomous vehicle merging protocol, in: QRS-C, pp. 333–342. doi:10.1109/QRS-C55045.2021.00057.

[17] M. Clavel, et al. (Ed.), 2007. All About Maude. volume 4350 of *LNCS*. Springer. doi:10.1007/978-3-540-71999-1.

[18] Mon, T.W., Bui, D.D., Tran, D.D., Ogata, K., 2021. Graphical animations of the ns(l)pk authentication protocols. JVLC 2021, 39–51. doi:10.18293/JVLC2021-N2-005.

[19] Nguyen, T.T.T., Ogata, K., 2017a. A Way to Comprehend Counterexamples Generated by the Maude LTL Model Checker, in: 2017

International Conference on Software Analysis, Testing and Evolution (SATE), pp. 53–62. doi:10.1109/SATE.2017.15.

[20] Nguyen, T.T.T., Ogata, K., 2017b. Graphical animations of state machines, in: 15th DASC, pp. 604–611. doi:10.1109/DASC-PICom-DataCom-CyberSciTec.2017.107.

[21] Robertson, G.G., Mackinlay, J.D., Card, S.K., 1991. Cone trees: animated 3d visualizations of hierarchical information, in: SIGCHI, pp. 189–194. doi:10.1145/108844.108883.

[22] Shor, P., 1994. Algorithms for quantum computation: discrete logarithms and factoring, in: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134. doi:10.1109/SFCS.1994.365700.

[23] Sugiyama, K., Tagawa, S., Toda, M., 1981. Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics 11, 109–125. doi:10.1109/TSMC.1981.4308636.

[24] Todorovic, D., 2008. Gestalt principles. Scholarpedia 3, 5345. doi:10.4249/scholarpedia.5345.

[25] Ware, C., 2012. Information Visualization: Perception for Design. 3 ed., Morgan Kaufmann.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Automating Leukemia Diagnosis with Autoencoders: A Comparative Study

Minoo Sayyadpour [a, *], Nasibe Moghaddamniya[b] and Touraj Banirostam[c]

[a]*Department of Math and Computer Science, Kharazmi University, Tehran, Iran*

[b]*Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*

[c]*Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran*

ARTICLE INFO

Article History:
Submitted 6.23.2023
Revised 7.28.2023
Accepted 8.1.2023

Keywords:
Leukemia
Deep learning
Auto-Encoder Neural Networks
Machine Learning

ABSTRACT

Leukemia is one of the most common and death-threatening types of cancer that threaten human life. Medical data from some of the patient's critical parameters contain valuable information hidden among these data. On this subject, deep learning can be used to extract this information. In this paper, AutoEncoders have been used to develop valuable features to help the precision of leukemia diagnosis. It has been attempted to get the best activation function and optimizer to use in AutoEncoder and designed the best architecture for this neural network. The proposed architecture is compared with this area's classical machine learning models. Our proposed method performs better than other machine learning in precision and f1-score metrics by more than 11%.

1. Introduction

Cancer is a pervasive ailment that transcends temporal and geographical boundaries. It stands as the second leading cause of mortality on a global scale and the third leading cause of death specifically within the nation of Iran. Projections indicate an anticipated escalation in the worldwide incidence rate of this affliction in the years ahead. Notably, leukemia occupies the fifth position in terms of prevalence worldwide and holds the second position within Iran [1]. Accurate and on-time recognition of this disease helps patients to be able to perform better treatments at earlier times and prevent the spread of cancerous cells and their spread to other organs [2]. Also, some types of cancer, such as leukemia, can be treated if detected on time. In the same way, increasing the accuracy of diagnosis can help save many people's lives. The process of recognizing cancer in the screening stage with methods such as CXR, mammography, sonography, endoscopy, and CT scan, compared to simpler blood and urine test methods, is expensive. Among the unknown aspects of cancer diagnosis, we can mention the following:

- How many samples are needed to train the detection system to achieve the desired accuracy, and to what extent can this data be accessed?
- Can other data be used with this technique to diagnose other diseases and other types of cancer?
- What are the parameters in the problem that are more important?
- To what extent can the doctor and the patient rely on these results?

In each sample (human), many parameters can probably be classified after some quantitative

*Corresponding author
Email address: std_minoosayyadpour@khu.ac.ir
ORCID: 0000-0001-8203-6442

DOI reference number: 10.18293/JVLC2023-N1-029

measurement to determine whether a person has cancer. However, finding compelling features in cancer diagnosis is a challenging process. Sometimes it is impossible to access an expert in the desired field (doctor, physician, etc.).

Deep learning automatically performs the feature extraction and learns which features are effective. This process is called Feature Extraction [2]. Deep learning can also perform classification based on features, and achieve higher accuracy in classification. These benefits have led to using deep learning in cancer diagnosis. Meanwhile, AutoEncoders can reduce features and extract efficient and valuable features; abstractly, they still maintain the main frame of the input sample. This led us to use Autoencoders to diagnose cancer. This paper has contributions that can be stated in the following:

- Using tabular data in AutoEncoders for cancer recognition.
- Use of Leukemia dataset of Sina Hospital in Hamadan.
- Find the best optimization algorithm and activation function for AutoEncoder networks.

The structure of remaining sections of this paper is organized as follows:

The second section reviews some essential methods in machine learning and deep learning networks. The third section deals with the research methodology. This section presents a new Leukemia detection method using AutoEncoder, showing the best results in accuracy and precision among the previous methods. The fourth chapter evaluates the results and compares them with each other. The fifth chapter concludes the study results.

2. Backgrounds

Neural networks can extract patterns and identify features difficult for humans to identify with their remarkable ability to derive results from complex data. This section discusses the studies conducted on the studies conducted in this field.

Medical Decision Support System (MDSS) helps doctors make decisions and accurately diagnose diseases. In 1980, MDSS was mentioned and brought about a revolution in decision support systems [3 and 4]. These systems help doctors to diagnose and track diseases and reduce possible errors. In [4], researchers designed a medical assistant system to help radiologists identify hylic tumors based on a distributed architecture with three specific nodes: Radiologist Visual Interface, Support services information system, and Web-based decision. Another medical assistant system to help asthma patients was presented by [5], which helps doctors control this chronic disease. In [6], they designed an MDSS to identify types of hearing problems, which helps experts control and solve them. In [7], another medical decision support system has been implemented to control Leukemia. In this system, in order to diagnose and predict Leukemia, four different classification methods have been used:

- Rule Based Reasoning
- Case-Based Reasoning
- Neural Network
- Discriminant Analysis

Many investigations have been done in diagnosing and classifying all types of cancer. Many of them use machine learning algorithms to classify data. In these references, in order to reduce the size of training data and define appropriate subsets of input variables, the feature extraction approach has been used. Feature extraction is the selection of a subset of input variables that increases the classifier's efficiency. In [8], they used a better diagnostic tool called Fine Needle Aspiration Cytology (FNAC) to classify the pattern of breast cancer. In this regard, Multivariate Adaptive Regression Splines (MARS) detect the appropriate subset of input variables from a neural network model. Therefore, the classification accuracy is increased in the presented hybrid methodology.

In [9], they used three models: Adaptive Resonance Theory Based Neural Network (ART), Self-Organizing Map Based Neural Network (SOM), and Back Propagation Neural Network (BPN) in order to classify types of breast cancer. Meanwhile, the BPN method performs better than the other two methods. BPN is one of the types of neural networks that works based on trial-and-error learning and tries to match the given inputs to the outputs by minimizing the value of an error function. Learning in this category of networks is supervised. ART is one of the types of unsupervised networks and is designed to allow the user to control the degree of similarity of the patterns placed in a cluster by adjusting the parameters of vigilance. SOM learning is also unsupervised and uses a competitive learning method for training. Its processing units are regularized in a competitive learning process for input patterns.

In [10], authors designed an expert system for breast cancer detection based on Association Rules (AR) and Neural Networks. Association rules have been used to reduce the dimensions of breast cancer database and Neural Networks for classification. In [11], different types of breast cancer have been classified by applying three methods of 4.5C tree, Multi-Layer Perceptron (MLP), and Naïve Bayes on specific markers of breast tumors. In order to classify

the types of Leukemia based on the analysis of gene expression data in [12] CBR learning method was used. In this method, previous examples are remembered, and past experiences are used when facing a new example. Diagnosis of Lymph Nodes Metastases (LNM) before surgery is challenging and complicated. [13], the artificial neural network method was used to diagnose this issue in patients with gastric cancer. In [14], the SVM method was used to classify breast cancer. Since the distinction between benign and malignant tumors is challenging and time-consuming, it seems necessary to use an automated method to detect these cases. In [15], a proposed method can detect and classify cancer from gene expression data using unsupervised deep learning methods. The main advantage of this method is the ability to use data from different types of cancer to automatically form features that help improve the identification and diagnosis of a specific type. In the proposed method, PCA is used to solve the problem of large dimensions of the raw feature space. The authors stated that applying this method to cancer data and comparing it with basic algorithms increased the accuracy of the cancer classification and provided a scalable and general approach to dealing with gene expression data.

In [16], a Sparse Stacked Autoencoder (SSAE) framework is used for automatic nucleus detection in cancer histopathology. The SSAE model can capture the high-level representational of point intensity without supervision. These high-level features have presented a classification that can effectively detect multiple nuclei from a large group of histopathological images. In this paper, the deep structures of SSAE are compared with other AutoEncoders in displaying the high-level features of the intensity of points. In [17], a layered feature selection method is presented along with SSAE. This paper used the Deep Learning method to classify tumors using gene expression data. With the help of SSAEs, high-level features have been extracted from the data. In each layer, a heuristic has been used to obtain relevant features to reduce the fine-tuning of calculations. Ultimately, the classifiers have used the data obtained in the features to perform tumor diagnosis. This paper has tested its presented method on 36 datasets. The datasets are from the GEMLeR source, which includes the gene expression data of 1545 samples from the cells of 9 types of tumors. In the evaluations, the presented method has better results in accuracy and ROC tests on GEMLeR.

In [18], the ability to use deep learning algorithms to detect lung cancer using LIDC images database has been investigated. Each image is divided into parts using the radiologist's marks. After reducing the sampling rate and rotating the images, 174,000 samples with a length and width of 52 pixels were obtained. CNN, DBN, and SDAE have been compared with other methods on a format of 28 image features and a support vector machine. The highest accuracy belongs to the DBN method (0.8119), and the accuracy of the CNN and SDAE methods is equal to 0.7976 and 0.7929, respectively. A vital point in this paper is the size of the images for lung cancer diagnosis, which can cause data loss if the sampling rate decreases. In [19], in 2016, a framework for early detection of prostate cancer from DW-MRI images was presented. The prostate is divided into parts in the first step based on a model trained by a Stochastic Speed function developed from NMF. NMF features are calculated using MRI intensity information, a probabilistic model, and interactions between prostate voxels. In the second step, ADC is obtained for different parts, its values are normalized, and CDF values are created for these prostate cells. In the last step, an AutoEncoder trained by an NCAE algorithm is used to classify prostate tumors as benign or malignant according to the CDFs extracted from the previous step.

In 2015, Deep Learning approaches were used to identify pathology in chest radiograph data [20]. Since large training sets are usually unavailable in the medical domain, the possibility of using deep learning methods based on non-medical training has also been investigated. The proposed algorithm has been tested on a set of 93 images, features extracted from CNN, and a set of features from non-medical domain obtained the best result. The proposed method has shown that deep learning with high-level non-medical image databases can be acceptable for general medical image recognition tasks. [21] introduces an ensemble hierarchical model for combining multiple classifiers. The proposed model consists of two steps: first, a Decision Tree and Logistic Regression models are trained independently, and then their outputs are fed into a Neural Network in the next level. The Neural Network is trained to combine the outputs of the previous classifiers, aiming to improve overall accuracy. The results showed that their proposed model achieved a classification accuracy of over 83%, which outperformed other existing methods in the literature.

In [22], a distributed Deep Learning framework is used to analyze and diagnose diseases. This analysis and diagnosis are based on the questions and answers of doctors and patients. Decision support systems are created using this information. Further, for diagnosis, distributed deep learning method is used to identify features and signatures.

3. Proposed Architecture

In this paper, the problem of diagnosis of Leukemia is a classification problem. In this kind of problem, feature extraction has particular importance because it increases data classification accuracy and makes it possible to classify data with large dimensions.

AutoEncoders are multilayer perceptron neural networks whose structure is not optimized for a specific form of data and can accept various types of data regardless of the local, temporal, and serial dependence between their features. AutoEncoders also provides the possibility of feature extraction process in unsupervised learning framework due to the existence of a structure including encoder and decoder. This structure helps the network to be trained in two phases, one phase for feature extraction and the second phase with the feature extractor classifier will be trained again with supervision, increasing the classification's accuracy.

To diagnose Leukemia from the data with the help of AutoEncoders, A structure including three steps of preprocessing, feature extraction, and classification is proposed. Accordingly, apart from machine learning methods used for comparison, this section mentions six experiments conducted to improve and increase classification accuracy. The proposed architecture is shown in Figure 1, and the input data size, activation functions, optimization algorithms, and the number of layers of AutoEncoder.

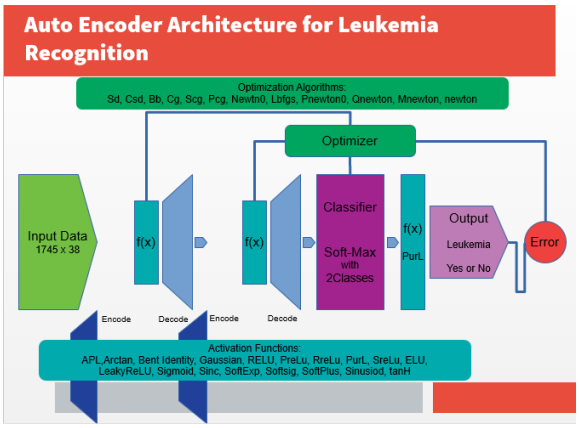


Figure 1: Proposed Architecture.

3.1 Preprocessing Data

Before performing any analysis on the data, it should be normalized, especially when the data is multidimensional. Using normalized data may have an inappropriate effect on the analysis results. Data normalization helps that their importance is independent of their measurement unit. The data were given a normal distribution between zero and one in the preprocessing unit before being given to AutoEncoder. In this test, the data distribution is normalized with the help of z-score standardization method.

3.2 Feature Extraction

The model trained in the first attempt had an input size of 38 features, and all the layers of AutoEncoder were given 100 epochs, which was unsupervised training. The whole model is also done with the classifier in 100 epochs for Fine Tune.

The AutoEncoder used in the first experiment has two groups of layers separated from the classifier. The AutoEncoder used in the first layer has 35 neurons, and in the second layer, it has 15 neurons (in the first layer group, the dimensions of the input data is 35 and then returns to the original space. In the second layer group, the input size is 35 will be taken from the hidden layer of the first group and then converted to 15).
The AutoEncoder used in the second experiment and all its attempts had two layers. It has 100 neurons in the first layer and 50 in the second.

The AutoEncoder used in the third experiment has 100 neurons in the first layer, 50 neurons in the second layer, and 25 neurons in the third layer.

The AutoEncoder has four layers in all attempts of the fourth experiment. The AutoEncoder used in the first layer has 100 neurons, the second layer has 50 neurons, the third layer has 25 neurons, and the fourth layer has 12 neurons.

The AutoEncoder used in all attempts of the fifth experiment has 50 neurons in four layers.

In the second test, the L2 weight regularization parameter for each of the network's three layers equals 0.0001. The value of this parameter was the same for all attempts made in the first experiment.

The sparsity proportion in the second experiment in the autoencoder networks is equal, and its value is 0.05. Softmax Regularization in the first test parameter equals 0.0001 for the classifier layer. Stack Regularization is equal to 0.0001 in the first experiment.

In each layer of Autoencoder, activation functions are used to activate each neuron, and parameters of the activation functions used in the structure of Autoencoder in the first experiment are non-adaptive. The activation functions used in the first test are Adaptive Piecewise Liner, arc tangent, Bent Identity, Gaussian, Rectified Liner Unit, Parametric Rectified Liner Unit, Randomized leaky Rectified linear unit, S-shaped Rectified Linear Activation Units, Exponential Liner, Leaky Rectified Linear Unit, sigmoid, sinc, Soft Exponential, Soft sign, Soft Plus, Sinusoid, and Hyperbolic Tangent.

The following optimization methods are used in the model used to train AutoEncoder and minimize the model error. These methods are Steepest Descent,

Cyclic Steepest Descent, Barzilai and Borwein Gradient, Nonlinear Conjugate Gradient, Scaled Nonlinear Conjugate Gradient, Preconditioned Nonlinear Conjugate Gradient, Hessian-Free Newton, Newton with Limited-Memory BFGS Updating Quasi, Preconditioned Hessian-Free Newton, Quasi-Newton Hessian approximation, and Newton's Method with Hessian Calculation every iteration.

3.3 Classifier

Softmax classifier was used to classify the data in the last layer. Also, the applied softmax is trained on 100 epochs and regularized at a rate of 0.001.

4. Simulation of Proposed Method

4.1 Dataset

The first issue in conducting this paper is to collect information on healthy people and patients with Leukemia. Therefore, to obtain the required data, Hamadan Ibn Sina Hospital was referred, and with the cooperation of this hospital, 1745 data samples were prepared and collected. This data sample is related to both genders, of which 55% are related to males, and 45% are related to females. The collected data contains 38 features: Age, gender, having an infection, lymphadenitis in the groin or Inguen (IN.LY), Fever and night sweats (FATH), muscle cramps, swelling of the liver or spleen (LSV), having Swoon, Loose skin, Hemorrhage, Pain bone, Anorexia (W.Irelish), Weight Loss (Depreciatory.bu), Nausea or puke, Cough or asthma, Numbness of the foot (IN.foot), leg swelling (WSF), White blood cell (WBC), Hematocrit, Hemoglobin count, The number of platelets, Sodium count, Lactate dehydrogenase enzyme (LDH), Uric acid, Cratinin, erythrocyte sedimentation rate (ESR), Prothrombin time (PT), PT activated (PTa), patrial thromboplastin time (PTT), Alkaline phosphatase (ALP), Bilirubin total (BillirobinT), Bilirubin direct (BillirobinD), glutamic-oxaloacetic transaminase (SGOT), Serum glutamic pyruvic transaminase (SGPT), mean corpuscular volume (MCV), Mean Corpuscular Hemoglobin (MCH), Mean corpuscular hemoglobin concentration (MCHC), Red blood cell distribution width (RDW), having cancer or not and type of cancer.

4.2 Implementation and Evaluation

The evaluation of the model's efficiency is divided into two parts. In the first phase, the layers of the Autoencoder are trained unsupervised, and the Root Mean Square Error (RMSE) is used as the error calculation criteria. In the second phase, the network is trained supervised, and a softmax classifier is placed at the end. The K-Fold cross-validation method and RMSE are used to evaluate the model. In the first test, the data is divided into five folds.

Based on the results obtained in the first experiment, it was found that the scg algorithm was the best optimization algorithm for training the Autoencoder, and the arctan function was the best activation function according to the optimization algorithm. The second objective was to find the best network structure. To compare the proposed model with other models, the evaluation criteria used were accuracy (ACC), sensitivity, precision, specificity, Matthew's correlation coefficient (MCC), F1-score, and Receiver Operating Characteristic (ROC).

Table 1: Top 10 implementations with higher Accuracy

Layers Neuron Size	Train Algorithm	Activation Function	Evaluation Criteria						
			Accuracy	*RMSE*	*Sensitivity*	*Specificity*	*Precision*	*MCC*	*F1-score*
[30, 15]	**scg**	**ArcTan**	**0.8297**	**0.1702**	**0.8546**	**0.8546**	**0.8546**	**0.8546**	**0.8546**
[30, 15]	pnewton0	BentIdentity	0.8286	0.1713	0.8500	0.7986	0.8559	0.6489	0.8525
[30, 15]	newton0	Softsig	0.8263	0.1736	0.8401	0.8068	0.8598	0.6457	0.8493
[30, 15]	pcg	Softsig	0.8229	0.1770	0.8431	0.7944	0.8525	0.6367	0.8476
[30, 15]	lbfgs	Softsig	0.8217	0.1782	0.8441	0.7903	0.8507	0.6345	0.8469
[30, 15]	cg	TanH	0.8206	0.1793	0.8450	0.7862	0.8481	0.6314	0.8463
[30, 15]	bb	TanH	0.8200	0.1799	0.8627	0.7600	0.8355	0.6282	0.8485
[30, 15]	lbfgs	ArcTan	0.8194	0.1805	0.8470	0.7806	0.8470	0.6310	0.8456
[30, 15]	pcg	ArcTan	0.8166	0.1833	0.8215	0.8096	0.8604	0.6292	0.8456
[30, 15]	newton0	Sigmoid	0.8166	0.1833	0.8333	0.7931	0.8504	0.6257	0.8411

According to Table 1, the best optimization algorithm for training is the scg, and the best activation function is ArcTan is selected and used in the following experiment. Then by changing the number of layers of Autoencoder in the following experiments, we want to find the best network structure.

Table 2: Results of AutoEncoder in Each Experiments

Layers Neuron Size	Evaluation Criteria						
	Accuracy	*RMSE*	*Sensitivity*	*Specificity*	*Precision*	*MCC*	*F1-score*
[100, 50]	0.8200	0.1799	0.8509	0.7765	0.8439	0.6320	0.8461
[100, 50, 25]	0.8240	0.1759	0.8441	0.7958	0.8542	0.6409	0.8482
[100, 50, 25, 12]	0.8246	0.1753	0.8539	0.7834	0.8486	0.6409	0.8502
[50, 50, 50, 50]	0.8246	0.1753	0.8480	0.7917	0.8520	0.6406	0.8494
[1024 512 250 100]	**0.8275**	**0.1724**	**0.8696**	**0.7682**	**0.8414**	**0.6442**	**0.8547**

After finding our architecture for AutoEncoder, we compare it to the classical machine learning method used in this area. According to the experiments conducted, AutoEncoder has a higher average accuracy among the classical methods, and the PCA-SVM method is in the next rank among the classical methods. These results are shown in Table 3 and Figure 2.

Table 3: Results of Comparing Proposed Method to Others.

Model	Evaluation Criteria	
	Accuracy	*F1-score*
PCA – 1NN	70.87%	73.46%
PCA – 3NN	69.64%	71.82%
PCA – 5NN	67.46%	69.88%
PCA – SVM	71.04%	73.96%
PCA - RBF	66.32%	68.03%
PCA - SoftMax	70.16%	72.78%
Proposed Model	**82.97%**	**85.68%**

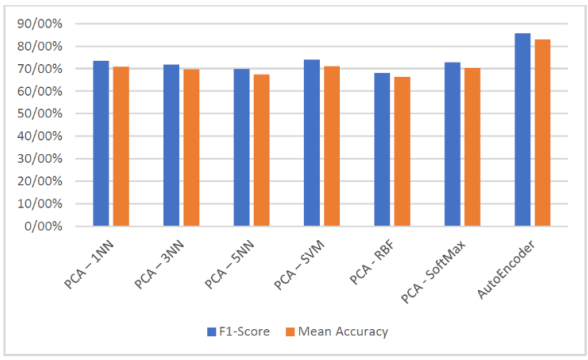


Figure 2: The results of machine learning and AutoEncoder model.

5. Conclusion

In Leukemia, cancer cells multiply rapidly, and the blood can no longer perform its functions. Early recognition of Leukemia helps patients to start treatment without interruption after diagnosis. In past research, machine learning methods have been used to diagnose cancer, each of which had average accuracy. With the introduction of AutoEncoders and increasing the accuracy of these networks, an attempt has been made to use them for Leukemia diagnosis. In this paper, the samples of the patients of Sina Hospital in Hamadan were used. Data preprocessing and normalization is the first step in using the dataset in this study. In this paper, the first question is to find the best optimization algorithm for training the weights of the AutoEncoder network, which was the SCG algorithm, and also the best activation function according to the optimization algorithm, which was ArcTan. The second step was finding the best structure of the network. Different tests by changing the layers of AutoEncoder resulted in two hidden layers: the number of neurons in the first was 30, and the number of neurons in the second was 15. The correctness of this network with the mean squared error of 0.17 has reached the average accuracy value of 82.97%, and its f1-Score is equal to 85.68%.

References

[1] Survival analysis of acute myeloid leukemia. RJMS 2015; 22 (134) :41-48.

[2] F. Yousefian, T. Banirostam, A. Azarkeivan, "Prediction Thalassemia Based on Artificial Intelligence Techniques: A Survey", Int. J. of Advanced Research in Computer and Communication Engineering, 2017, Volume 6, Issue 8, Pages 281-287

[3] K.V. Rodpysh, S.J. Mirabedini, T. Banirostam, "Resolving cold start and sparse data challenge in recommender systems using multi-level singular value decomposition," Computers & Electrical Engineering, Volume 94, September 2021, 107361.

[4] J. M. Garcia-Gomez, C. Vidal, J. Vicente, L. Marti-Bonmati, and M. Robles, "Medical Decision Support System for Diagnosis of Soft Tissue Tumors based on Distributed Architecture," in The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2004, vol. 4, pp. 3225–3228.

[5] M. H. Hamza, A medical decision support system for asthmatic patient health care. Anaheim [u.a.]: International Assoc Acta Press, 2002.

[6] A. Abu Bakar, Z. Othman, R. Ismail, and Z. Zakari, "Using rough set theory for mining the level of hearing loss diagnosis knowledge," in 2009 International Conference on Electrical Engineering and Informatics, 2009, pp. 7–11.

[7] Y. M. Chae, K. S. Park, Q. Park, and M. Y. Bae, "Development of medical decision support system for leukemia management.," Stud. Health Technol. Inform., vol. 52 Pt 1, pp. 449–52, 1998.

[8] S.-M. Chou, T.-S. Lee, Y. E. Shao, and I.-F. Chen, "Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines," Expert Syst. Appl., vol. 27, no. 1, pp. 133–142, Jul. 2004.

[9] K. Mumtaz, S. A. Sheriff, and K. Duraiswamy, Evaluation of three neural network models using Wisconsin breast cancer database. International Conference on Control, Automation, Communication and Energy Conservation, 2009.

[10] M. Karabatak and M. C. Ince, "An expert system for detection of breast cancer based on association rules and neural network," Expert Syst. Appl., vol. 36, no. 2, pp. 3465–3469, Mar. 2009.

[11] D. Soria, J. M. Garibaldi, E. Biganzoli, and I. O. Ellis, "A Comparison of Three Different Methods for Classification of Breast Cancer Data," in 2008 Seventh International Conference on Machine Learning and Applications, 2008, pp. 619–624.

[12] J. F. De Paz, S. Rodríguez, J. Bajo, and J. M. Corchado, "CBR System for Diagnosis of Patients," in 2008 Eighth International Conference on Hybrid Intelligent Systems, 2008, pp. 807–812.

[13] E. H. Bollschweiler, S. P. Monig, K. Hensler, S. E. Baldus, K. Maruyama, and A. H. Helscher, "Artificial Neural Network for Prediction of Lymph Node Metastases in Gastric Cancer: A Phase II Diagnostic Study," Ann. Surg. Oncol., vol. 11, no. 5, pp. 506–511, May 2004.

[14] M. Sewak, P. Vaidya, C.-C. Chan, and Zhong-Hui Duan, "SVM Approach to Breast Cancer Classification," in Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007), 2007, pp. 32–37.

[15] R. Fakoor, F. Ladhak, … A. N.-P. of the, and U. 2013, "Using deep learning to enhance cancer diagnosis and classification," dl.matlabproject.ir, 2013.

[16] J. Xu et al., "Stacked Sparse Autoencoder (SSAE) for Nuclei Detection on Breast Cancer Histopathology Images," IEEE Trans. Med. Imaging, vol. 35, no. 1, pp. 119–130, Jan. 2016.

[17] V. Singh, N. Baranwal, R. K. Sevakula, N. K. Verma, and Y. Cui, "Layerwise feature selection in Stacked Sparse Auto-Encoder for tumor type prediction," in 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2016, pp. 1542–1548.

[18] W. Sun, B. Zheng, and W. Qian, "Computer aided lung cancer diagnosis with deep learning algorithms," 2016, vol. 9785, p. 97850Z.

[19] I. Reda et al., "A new NMF-autoencoder based CAD system for early diagnosis of prostate cancer," in 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), 2016, pp. 1237–1240.

[20] Y. Bar, I. Diamant, L. Wolf, and H. Greenspan, "Deep learning with non-medical training used for chest pathology identification," 2015, p. 94140V.

[21] M. Abedini, A. Bijari and T. Banirostam, "Classification of Pima Indian Diabetes Dataset using Ensemble of Decision Tree, Logistic Regression and Neural Network," International Journal of Advanced Research in Computer and Communication Engineering, Vol. 9, Issue 7, 2020, pp. 1-4.

[22] Ms. K. Remya, Dr. T. Senthil Prakash, Mr. S. Ramesh, and Ms.P.C. Saritha, "Sparse Deep Learning Framework for Disease Analysis," IRETS - Int. Journals, vol. 2, 2015.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

A Topic Lifecycle Trend Prediction Algorithm on Facebook

Chen Luo[a,*], Jun Shi[a]

[a]CyberAray Network Technology Co.,Ltd, China

ARTICLE INFO

Article History:
Submitted 5.6.2023
Revised 7.18.2023
Second Revision 7.28.2023
Accepted 7.30.2023

Keywords:
Opinions
Facebook
Hot topics
Trend prediction
Similarity
Lifecycle

ABSTRACT

Recently, social media has been widely used for people to discuss public opinions and share their views. Internet public opinions have attracted a lot of attention from the government, enterprises and the general public. How to properly analyze, utilize and guide these online opinions is an extremely important issue that the world is currently faced with, and the prediction of topic lifecycle trends is the key to solving this problem. This paper proposes a topic lifecycle trend prediction algorithm based on Facebook data. The algorithm takes into account the similarity between new topics and historical topics in terms of lifecycle curves and the similarity in terms of text content, finds a curve that can best represent the future lifecycle trend of new topics, and then effectively predicts the trend of new topics. It is helpful and meaningful to use this method in the early warnings and predictions of online public opinions and hot topics.

1. Introduction

Nowadays, social media has become an indispensable part of people's daily lives. According to the Digital 2020 July Global Statshot report released by We Are Social and Hootsuite [1], we know that there are now 3.96 billion social media users worldwide, accounting for about 51% of the world's total population. Therefore, social media is one of the most important ways to propagate and spread information. In those various social media platforms, there are many hot topics generated every day. These hot topics are significant carriers which contain focused information people pay attention to, and they are directly related to the size of the social influence triggered by the events. It is crucial to make good use of the information in hot topics. On one hand, the government can monitor and analyze hot topics to understand the trend of online public opinions and take corresponding measures in time, which is conducive to maintaining the long-term stability of society. On the other hand, enterprises can understand the needs of users through relevant hot topics to make business plans such as personalized marketing to some users. As a result, putting forward a method that can predict the trend and analyze the lifecycle of topics is of great importance.

Facebook, as the world's most popular social media platform with over 2.6 billion monthly active users, has a plenty of topic data. In this paper, we use posts information from Facebook as our datasets, which include date, time, content and some other useful information. We first extract daily hot topics from Facebook daily posts, and then we use Jaccard Similarity algorithm to calculate the similarities between daily hot topics and posts from other days by comparing their keywords in order to find those posts which are related to hot topics. Based on that, we use the number of relevant posts as the value of y-axis, dates as the value of x-axis to plot the lifecycle curve of each hot topic. Topics with similar lifecycle curves are merged into one cluster, which means all topics under the same cluster have similar trends. After that, we extract a centroid curve for each cluster to stand for the trend of the cluster. When a new topic comes, Dynamic Time Warping (DTW) algorithm is used to compute similarities between curves to find curves from all clusters that are most similar to the curve of new topic. From those topics which are similar in curve, we check if their contents are similar to the new topic as well. Considering similarity both in curve and in content, we

*Corresponding author
Email address: linc950412@gmail.com
ORCID: 0000-0002-5747-742X

DOI reference number: 10.18293/JVLC2023-N1-30

can find one curve that best represents the future lifecycle trend of a new topic.

On the basis of description above, this paper proposes a topic lifecycle trend prediction algorithm based on Facebook dataset. There is no doubt that we encountered many difficulties in this process, such as finding posts that are related to hot topics, complicate data cleaning, reducing the time complexity of algorithm operation to increase efficiency and so on. With the efforts of keeping trying and optimizing, we finally propose this method. The main contribution of this paper can be summarized as follows:

- We use the K-Shape algorithm to cluster time series data, making topics with similar lifecycle curves into one cluster, which allows us to effectively observe and analyze the characteristics of different types of topic lifecycle, and make effective trend predictions for new topics that have similar curve characteristics to historical ones.
- We use the DTW algorithm to calculate the similarity between new topics and historical topics on the curve, solving the problem that ordinary Euclidean distance cannot compare the similarity of two unaligned or unequal length sequences.
- Considering the similarity of topics both in lifecycle curve and in text content, we propose a method to predict topic lifecycle trend.

The rest of the paper is discussed as the following order, Chapter 2 introduces the related work. In Chapter 3, we present the topic lifecycle trend prediction method. Chapter 4 shows the experiments and evaluation. At last, conclusion and future work is discussed in Chapter 5.

2. Related Work

In 2007, Jiang, Yue [2] made a study showing that early lifecycle data can be used to predict the fault-prone modules in a project. In 2012, Shota Ishikawa [3] designed a system detecting hot topics during a certain period of time and a method was proposed to reduce the variation of posted words related to the same topic, which provides a great contribution to AI services. In the same year, Rong Lu and Qing Yang [4] defined a new concept as trend momentum, which are used to predict the trend of news topics. Juanjuan Zhao [5] developed a model of short-term trend prediction of topics based on Sina Weibo dataset while the accuracy still needs to be improved when the trend of topic changes too frequently in 2014. More recently in 2018, Abuhay [6] used NMF topic modeling method to find topics and implemented ARIMA to forecast the trend of research topics. Chaoyang Chen and Zhitao Wang [7] proposed a correlated neural influence model, which can predict the trending research topics among the research evolution of mutually influenced conferences in the same year. In 2021, Yumei Liu and Shuai Zhang [8] researched the use of blockchain technology in the financial field, utilized various kinds of methods like co-word analysis and bi-cluster algorithm to explore hot topics and predict the future development trend. In 2022, a scientific research topic trend prediction model based on multi-LSTM and Graph Convolutional Network was proposed by Mingying Xu and Junping Du [9]. Compared to other baseline models, its experiment results showed an improvement on the prediction precision.

3. Topic Lifecycle Trend Prediction Method

Our goal is to build a topic lifecycle trend prediction algorithm model. Before that there are some necessary work to be done first, including hot topic extraction, finding topic-related posts, drawing historical topic lifecycle graphs, clustering and curve fitting based on lifecycle curve shapes. After all these tasks are completed, we calculate the shape similarity between topic lifecycle curves by using DTW algorithm. In the meantime, we calculate text content similarity based on keyword matching. Considering both shape similarity and text content similarity, we propose a topic lifecycle prediction method to predict the future lifecycle of new topics. The detailed flowchart is shown in Fig 1.

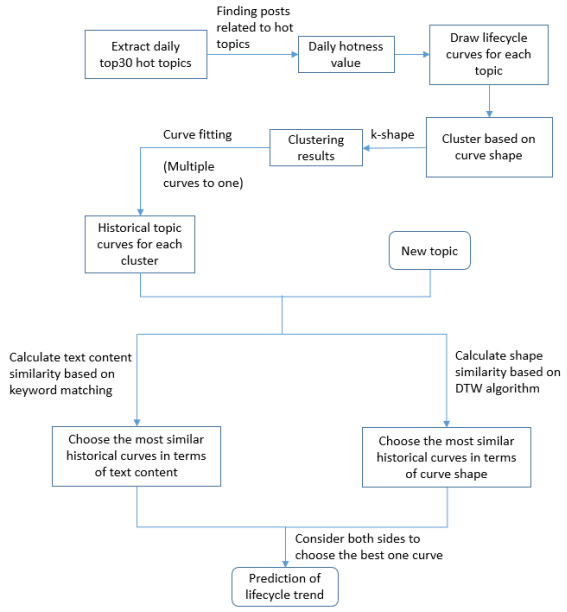


Figure 1: Flow chart of topic trend prediction method.

3.1 Hot Topic Extraction

In this paper, the data we used came from crawling the Facebook platform. We crawled some of the posts' information of the Facebook platform from May 2020 to April 2021 as needed, a total of 100,535,793 posts with non-empty content.

Since there are too many post contents in different languages, we cannot analyze all of them. Then we intend to study the posts' information of only two languages this time, English and Chinese, by considering the number of language users, language popularity and some other factors. Thus, the first step is to identify Chinese and English posts, and then we perform tokenization. For Chinese lexical analysis, we use the tool called HanLP, and for foreign language

lexical analysis, we use spaCy. Both HanLP and spaCy are commonly used natural language processing tools. Subsequently, keyword extraction is performed. A combination of lexical analysis, entity recognition, TF-IDF [10] and TextRank [11] algorithms are used to extract keywords, which can be classified by category as people, places, organizations, etc. Then the single pass algorithm is used to cluster the sentences in the post content based on keyword similarity, and the sentences with similar keywords are clustered into one class, that is, into one topic. Next, we need to give each topic a topic description to stand for the content of this topic. By achieving that, we extract the keywords of this topic, calculate the similarity between the keywords of the topic and the keywords of each sentence in the topic in order, and select the sentence with the highest similarity score and the shortest length as the topic description of this topic. Besides, we need to know each topic's hotness to find hot topics. To reach this goal, the number of posts of each topic is taken as the topic's hotness. We choose the daily Top N topics with highest hotness value as the main topics of this study. Due to the presence of some advertising information in the extracted hot topics, which are useless, it is experimentally concluded that when N=30 is chosen (N is not unique), a sufficient number of valid topics can be guaranteed. In this case, we extract the daily Top 30 hot topics for this study, and there are 9900 topics in total for eleven months. Because there is a possibility that a topic may be a hot topic for several days in a row, we are supposed to de-duplicate these 9900 topics and finally get 8359 unduplicated topics, which are used as our historical topic library (including curves and text contents) for this study. These topics are like "President Donald Trump announced that he and first lady Melania Trump has tested positive for COVID-19", "Joe Biden just overtook Donald Trump in Pennsylvania, where he's now leading by 5,594 votes".

3.2 Finding Topic-Related Posts

In In order to study the lifecycle curve of a topic, we need to know how hot the topic is on a daily basis. Thus, we need an algorithm to find the statistics of the number of posts a topic has on a daily basis, as the daily hotness of the topic.

By achieving this goal, we design a Topic-Finding-Posts algorithm. The algorithm performs keyword extraction from a library of posts to be searched to obtain keywords for each post, and then it calculates the Jaccard similarity coefficient score between the set of post keywords and the set of topic keywords to see if the post is similar to the topic or not. The higher the score, the more similar the two sets. Finally, it outputs the posts related to the topic.

Due to the sheer volume of computing and the limitations of machine, we are unable to find posts related to a topic for an entire year. Given that hot topics are generally not hotter than three months, we set the lifecycle to three months, the month in which the topic is located, the month before and the month after. For example, if a hot topic is on July 3, 2020, we would look for posts related to the topic in June, July and August of 2020, which means it requires us to calculate the Jaccard similarity coefficient score between the set of post keywords in these three months and the set of topic keywords to find topic-related posts. The flow chart of this algorithm is shown in Fig. 2.

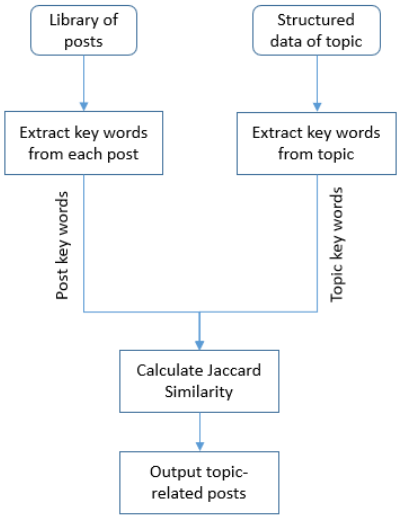


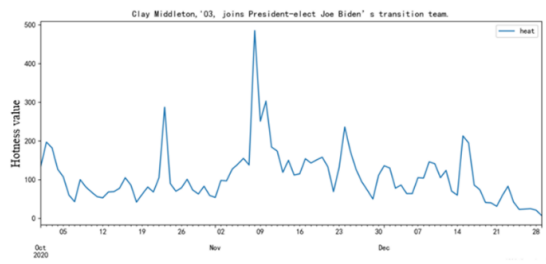
Figure 2: Flow chart of Topic-Finding-Posts algorithm.

3.3 Drawing Historical Topic Lifecycle Graphs

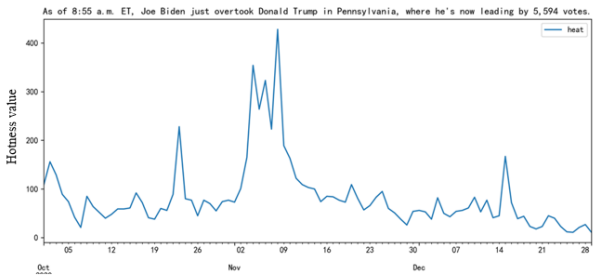
This step is to draw lifecycle graphs of all historical topics to build the historical topic library needed for this study. According to Part 3.1, we know that there are a total of 8359 unique topics. The Topic-Finding-Posts algorithm of Part 3.2 is used to find posts related to these topics over a three-month period and the number is counted as the topic's hotness value. Using the topic's hotness value as the y-axis, the three-month span of time as the x-axis, and the topic description as the title, the lifecycle graphs of these topics are plotted and saved as a historical topic graph library, representing the lifecycle trends of hot topics that emerged during these eleven months.

3.4 Topic Lifecycle Curve Shape Clustering Based on K-Shape

Based on the results of Part 3.3, we can obtain lifecycle graphs of thousands of historical topics. Since several hot topics appearing at the same time may be discussing the same thing, from this perspective they are actually one topic. For example, topic "Clay Middleton, '03, joins President-elect Joe Biden's transition team" and topic "Joe Biden just overtook Donald Trump in Pennsylvania, where he's now leading by 5,594 votes" appeared in November 2020 are both about the US election, and they are similar in terms of their lifecycle curves. These two topics' lifecycle curves are shown in Fig 3.



(a) Lifecycle curve of topic "Clay Middleton, '03, joins President-elect Joe Biden's transition team".



(b) Lifecycle curve of topic "Joe Biden just overtook Donald Trump in Pennsylvania, where he's now leading by 5,594 votes".

Figure 3: Examples of two topics that have similar lifecycle curves.

In response to this case, we decide to cluster topics with similar lifecycle curves into one class and form a curve that represents this class as the lifecycle curve for this class. There are several advantages of doing this. First of all, it reduces the amount of computation since we only take the curve that represents each cluster into account. Secondly, it may help reduce the errors caused by the Topic-Finding-Posts algorithm on the hotness value of the topic lifecycle graph. Thirdly, it can eliminate some noise. For example, some oddly shaped graphs that appear only once are not representative, indicating that they are not common hot topics and will most likely not appear again in the future, which is not helpful for prediction, and these outlier topics can be found and discarded through clustering.

In this paper, K-Shape algorithm [12] is used for clustering, which is a clustering algorithm specifically for time series data and is concerned with similarity of shape. We use the tslearn package for clustering, which requires that the lengths of the different sequences should be the same. Thus, we cluster the curves by month, which ensures that the time series lengths of the lifecycle curves of topics in the same month are the same. In addition, we need to do feature scaling to bring all hotness values to the same magnitudes. To do this, we standardize the time series data by using z-normalization. Then, we use the normalized dataset to perform K-Shape clustering, where similar curves are clustered in one class and output centroid curves representing the lifecycle curves in this class. For example, the above lifecycle curves in Fig 3 are similar and can be clustered into one class, whose centroid curve is shown in Fig 4.

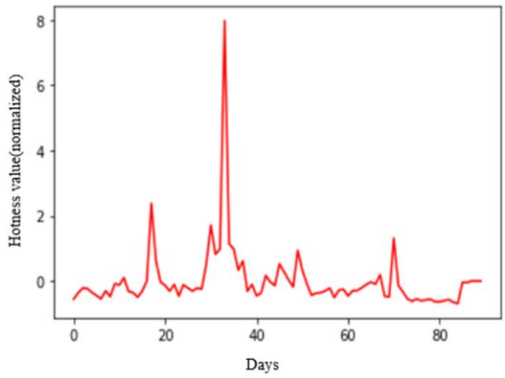


Figure 4: Example of centroid curve of a cluster (z-normalization).

All centroid curves after clustering are collected as a historical topic graph library. When a new topic emerges, the shape similarity between the new topic and the curves in historical topic graph library can be compared to predict the future trend of the new topic.

3.5 Calculation of Shape Similarity Based on DTW

DTW (Dynamic Time Warping) [13] is a dynamic programming algorithm that calculates the similarity of two time series [14], especially those of different lengths. When a new topic has been around for a while, we use this algorithm to calculate the shape similarity between the lifecycle curve of the new topic at that point and the centroid curves in the historical topic graph library in turn, and rank them to get some historical topic curves that are most similar to the current new topic. This gives an indication of some possible future trend directions for the new topic.

3.6 Calculation of Text Content Similarity Based on Keyword Matching

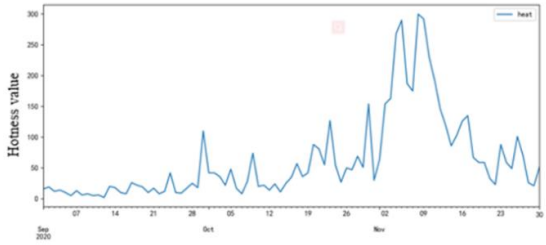
From Part 3.4, we can see that we have successfully clustered topics with similar lifecycle curves and obtained the centroid curves representing each category of topics. Next, we perform keyword extraction for each category of topics to learn the main textual content. The Jaccard similarity coefficient score between these topic keywords and the new topic keywords are calculated in turn and ranked to find the curves of historical topics that are most similar to the new topic in terms of textual content. This gives an indication of some possible future trend directions for new topics when considering similarity in text content. When a new topic has just come out and there is no obvious curve, text content similarity can be considered to use to solve the cold start problem, but only for reference, as similar text content does not mean that the trend is similar.

4. Experiments and Evaluation

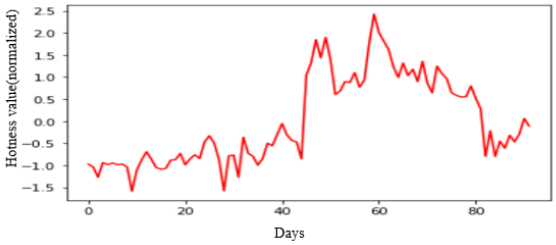
4.1 Topic Trend Prediction

Following the order in Chapter 3, we first extract the

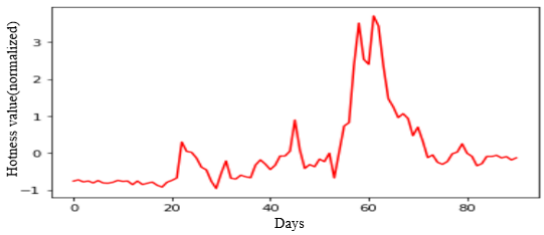
daily hot topics from the training set, then use the Topic-Finding-Posts algorithm to find the topic-related posts in a three-month cycle (here we set the similarity threshold of 0.45, if the Jaccard similarity coefficient score is greater than the threshold, the post is considered relevant to the topic). After that, we count the number of daily posts as the hotness to draw the lifecycle curve of the topic, and similar curves are clustered. Next, the DTW-based shape similarity calculation is performed on the clustered centroid curves and the test topic curves, and it is tested that when the similarity distance is less than 3.4, the trends of the two topics are similar. At the same time, the text content similarity calculation based on keyword matching is also performed (here the similarity threshold is also 0.45, and when the similarity is greater than 0.45, the text contents of the two topics are similar). Based on these experiments, historical topics that meet all the above conditions are considered similar to new topics, and their lifecycle trends can be used as a prediction of future trends of new topics. Let's take an example (see Fig 5), when the test topic "Trump's dream is America's dream, Biden's dream is China's dream, Ivanka says" shows a lifecycle trend (see Fig 5. a, the part of the diagram within the dotted line), we can get three similar curves from historical topic graph library by only considering shape similarity (see Fig 5. b c d). Then we consider text content similarity, only one curve meets all the conditions at the same time, which is the second one of the three predictions (see Fig 5. c). Then we get our predicted trend curve for the test topic.



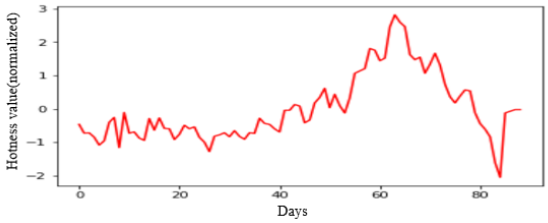
(a)



(b)



(c) the right one



(d)

Figure 5: Topic trend prediction experiments.

4.2 Clustering Effect Evaluation – Silhouette Coefficient Prediction

We use the Silhouette Coefficient as the effect evaluation index of this K-Shape clustering. The Silhouette Coefficient is a useful metric for evaluating clustering performance. It is computed by using mean distance between data points in the same cluster (cohesion) compared to the mean distance between data points in other clusters (separation) [15]. The calculated score ranges from -1.0 to 1.0. The higher the score, the better the clustering effect. To make the score computable, there have to be at least two clusters.

Assume that the data have been clustered into *k* classes. For data point $x(i) \in K$ (*K* is the cluster containing all the data points $x(i)$), $a_{x(i)}$ is the mean distance between $x(i)$ and every data point in the cluster *K*, $b_{x(i)}$ is the minimum mean distance between $x(i)$ and every data point in other clusters that is not a member of *K*. The calculation [16] of the Silhouette Coefficient of $x(i)$, the Silhouette Coefficient of each cluster, and the Silhouette Coefficient of all clusters can be shown as in (1), (2), and (3), respectively.

$$S_{(x_i)} = \frac{b_{x(i)} - a_{x(i)}}{max(a_{(x(i)}, b_{x(i)})} \quad (1)$$

where

$x(i)$ = data point in the cluster, $i = 1, 2, 3, \ldots, n$,

$a_{x(i)}$ = the mean distance between $x(i)$ and every data point in the cluster *K*, and

$b_{x(i)}$ = the minimum mean distance between $x(i)$ and every data point in other clusters that is not a member of *K*.

$$S_m = \frac{1}{n}\sum_{i=1}^{n} S_{(x_i)} \quad (2)$$

where

m = the number of the cluster, and

n = the number of data points in the same cluster.

$$S_{avg} = \frac{1}{k}\sum_{m=1}^{k} S_m \quad (3)$$

where

k = number of all clusters.

We take the data of November 2020 as an example and cluster out ten classes as shown below (see Fig 6), where the red line represents the centroid curve of each class with a S_{avg} of 0.5162703634739744. The results tell us that the clustering works well. Data from other months are treated in the same way.

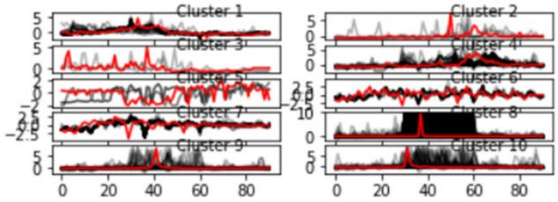


Figure 6: Clustering results of the data of Nov 2020(normalized).

4.3 Prediction Effect Evaluation

According to the clustering results of Part 4.2, we can briefly classify the type of clustering into short lifecycle class topics and long lifecycle class topics. The short ones refer to as above cluster 8, 9, 10, suddenly appeared to reach the peak and then the hotness immediately dropped and disappeared, mostly for some sudden events whose whole duration is just a few days. While the long ones are like cluster 5, 6, 7, whose hotness duration is long enough. They are usually serious events that need to be widely discussed. In this paper, we use the data of test set (120 topics in Jul 2021) as a collection of new topics. This method performs well on the test set, and the accuracy can reach 90%. For those test topics that are incorrectly predicted, we can see that there has not been a curve in the historical topic graph library similar to the curve of these test topics and there are no similar keywords in the text content either. In response to this situation, we add the lifecycle curves of these incorrectly predicted topics into the historical topic graph library to enrich it, so that these unmatched curves can be matched in the future.

5. Conclusion and Future Work

This paper proposes a topic lifecycle trend prediction algorithm based on Facebook data, which integrates shape similarity and text content similarity, and the results are more accurate than considering only shape similarity or only text content. The experimental results also show that the method is effective, but there are still some shortcomings that need to be improved.

1. For topics or lifecycle curves that have not appeared in history, it is impossible to make effective predictions, and the possible solution is to continuously expand the historical topic library to cover as many topics and curves as possible.

2. Because of the large amount of data, some topics have a very long lifecycle and the complete curve cannot be obtained. The algorithm can be optimized later to improve the running speed so that the complete lifecycle curve can be reached.

3. For shape similarity, the new topic needs to have a long enough lifecycle curve (to cover local or global features) to determine whether two topics are really similar by shape similarity calculation.

Acknowledgment

I would like to express my great appreciation and thanks to my supervisor Dr. Jun Shi, for her patient guidance and useful suggestions. I would also like to thank all my colleagues, for brainstorming with me and helping me solve the problems I encountered in this work. Finally, special thanks should be given to Shenzhen CyberAray Network Technology Co., Ltd, for supporting the data and resources I need.

References

[1] Kemp S. Digital 2020: July Global Statshot, Datareeportal.

[2] Jiang, Yue, Bojan Cukic, and Tim Menzies. "Fault prediction using early lifecycle data." The 18th IEEE International Symposium on Software Reliability (ISSRE'07). IEEE, 2007.

[3] Ishikawa, Shota, et al. "Hot topic detection in local areas using Twitter and Wikipedia." ARCS 2012. IEEE, 2012.

[4] Lu, Rong, and Qing Yang. "Trend analysis of news topics on twitter." International Journal of Machine Learning and Computing 2.3 (2012): 327.

[5] Zhao, Juanjuan, et al. "A short-term trend prediction model of topic over Sina Weibo dataset." Journal of Combinatorial Optimization 28 (2014): 613-625.

[6] Abuhay, Tesfamariam M., Yemisrach G. Nigatie, and Sergey V. Kovalchuk. "Towards predicting trend of scientific research topics using topic modeling." Procedia Computer Science 136 (2018): 304-310.

[7] Chen, Chengyao, et al. "Modeling scientific influence for research trending topic prediction." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.

[8] Liu, Yunmei, et al. "The sustainable development of financial topic detection and trend prediction by data mining." Sustainability 13.14 (2021): 7585.

[9] Xu, Mingying, et al. "A scientific research topic trend prediction model based on multi-LSTM and graph convolutional network." International Journal of Intelligent Systems 37.9 (2022): 6331-6353.

[10] Ramos, Juan. "Using tf-idf to determine word relevance in document queries." Proceedings of the first instructional conference on machine learning. Vol. 242. No. 1. 2003.

[11] Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.

[12] Paparrizos, John, and Luis Gravano. "k-shape: Efficient and accurate clustering of time series." Proceedings of the 2015 ACM SIGMOD international conference on management of data. 2015.

[13] Myers, Cory, Lawrence Rabiner, and Aaron Rosenberg. "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition." IEEE Transactions on Acoustics, Speech, and Signal Processing 28.6 (1980): 623-635.

[14] Meesrikamolkul, Warissara, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. "Shape-based clustering for time series data." Advances in Knowledge Discovery and Data Mining: 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Proceedings, Part I 16. Springer Berlin Heidelberg, 2012.

[15] Kaoungku, Nuntawut, et al. "The silhouette width criterion for clustering and association mining to select image features." International journal of machine learning and computing 8.1 (2018): 69-73.

[16] Aranganayagi, S., and Kuttiyannan Thangavel. "Clustering categorical data using silhouette coefficient as a relocating measure." International conference on computational intelligence and multimedia applications (ICCIMA 2007). Vol. 2. IEEE, 2007.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Enhanced Emotion Detection and Analysis in Human-Robot Interactions: An Innovative Model and Its Experimental Validation

Alfredo Cuzzocrea[a,b,*], Alessia Fantini[c,d] and Giovanni Pilato[d]

[a] iDEA Lab, University of Calabria, Rende, Italy
[b] Dept. of Computer Science, University of Paris City, Paris, France
[c] University of Pisa, Pisa, Italy
[d] CNR, Istituto di Calcolo e Reti ad Alte Prestazioni, Consiglio Nazionale delle Ricerche, Palermo, Italy

ARTICLE INFO

Article History:
Submitted: 7.1.2023
Revised: 7.26.2023
Accepted : 7.28.2023

Keywords:
Emotion Detection
Human-Robot Interaction
ChatGPT
Dialogues Generation

ABSTRACT

This study presents an ongoing project on developing an Italian textual dataset for emotion recognition in human-robot interaction (HRI). The main goal is constructing a dataset using a well-defined methodology that generates custom interactions. To accomplish this, we employed ChatGPT to assist us in developing the dialogues, which human psychology experts then reviewed. Our analysis primarily focused on assessing various aspects of the dataset, including the distribution of context types, gender representation, consistency between context and emotion, and the quality of interaction. By "quality," we refer to how the generated text accurately reflects the intended manifestation of emotions. Based on the analysis, we modified the dialogues to emphasize specific emotions in particular contexts. The findings from this preliminary study were significant, providing valuable insights to guide the generation of subsequent conversations and facilitate the creation of a more comprehensive dataset. A case study is also outlined with the aim of enabling increasingly realistic interactions in HRI scenarios.

1. Introduction

Emotions play a crucial role in Human-Robot Interaction (HRI), but one of the most challenging tasks for robots in such interactions is recognizing emotions [47], [19]. Emotions are multidimensional, and their comprehension relies on the context in which they are expressed. Understanding emotions necessitates considering context, which challenges Natural Language Processing (NLP) research. Context enables us to predict emotions to some extent. For instance, attending a party, securing a new job, or embarking on a trip are highly likely to evoke the feeling of "joy." Conversely, experiencing a loss or arguing with a loved one is often associated with “sadness.”

While emotions can overlap and vary among individuals, certain objective situations tend to be consistently linked to specific emotions. Therefore, providing examples of context-related emotions can aid in identifying emotions accurately. In a study on conversational context modeling [48], the authors highlight the significant enhancement that considering context can bring to NLP systems. Consequently, constructing a specific and rich contextual information dataset is vital within data-driven models.

A relevant number of contributions in the literature focus on developing datasets for emotion recognition. However, the majority of these datasets typically encompass only a limited number of emotions, often centered around Ekman’s basic emotions. Some examples are EmotionX [54] , Affect-Intensity Lexicon and Emotion Dataset (AILA) [42], Crowd-Flower’s Emotion Dataset [1], Friends [30] , EmoBank [11]. Furthermore, many approaches build dataset using news paper, books or dialogues found on the Internet, including those found from social media, e.g. SemEval-2018 Task 1: Affect in Tweets (AIT-2018) [43], Sentiment140 [25], Emotion Intensity Dataset (EmoInt) [41], The International Survey on Emotion Antecedents and Reactions (ISEAR) [52]. Others use movies, e.g. The Stanford Sentiment and Emo-

*This research has been made in the context of the Excellence Chair in Big Data Management and Analytics at University of Paris City, Paris, France

alfredo.cuzzocrea@unical.it (A. Cuzzocrea); alessia.fantini@phd.unipi.it (A. Fantini); giovanni.pilato@icar.cnr.it (G. Pilato)
ORCID(s): 0000-0002-7104-6415 (A. Cuzzocrea); 0009-0007-0337-2423 (A. Fantini); 0000-0002-6254-2249 (G. Pilato)

DOI reference number: 10.18293/JVLC2023-N1-031

tion Classification (SSEC) [53, 44] or physiological signals, e.g. The DEAP (Database for Emotion Analysis using Physiological Signals) [33]. In addition to this, the general problem has been stirred-up by several research efforts in the community (e.g., [58, 12, 40, 57, 32]).

Regarding Italian datasets, there are fewer contributions and often from tweets, some of the most widely used include SEMEVAL-ITA-2018 [13], ITA-EVALITA-2020 [7], EmoLexIta [16], The STS-ITA (Sentences in the Wild - Italian) [9], or news articles e.g. News-ITA [50]. A lexicon based approach has been also used for sentiment classification of books reviews in the Italian language [15].

Regarding the primary contributions in the existing literature, we intentionally chose to exclude data from social media or newspaper articles. Such content often exhibits language that may not align well with natural interactions. Instead, for our specific usage scenarios, such as Human-Robot Interactions, we utilized examples of interactions between individuals that emphasize the emotions we wish to focus on. This particular aspect holds significant importance in our study because the dialogue structure enables us to furnish the robot with examples of interactions closely resembling those that occur in real-world settings. Through the generation of custom dialogues, we were able to provide precise contextual scenarios in which particular emotions might arise. Also, the labeling was not done directly by us: this is another of the challenges highlighted by [48] in conversational context. We asked the ChatGPT to generate dialogues in which a specific emotion, such as *joy*, emerges; subsequently, we monitored and possibly adjusted or validated the associated labeling. An additional crucial aspect of our research is that we go beyond merely incorporating basic emotions; we have identified and labeled a comprehensive set of fourteen emotions, considering those likely to arise during Human-Robot Interaction (HRI) across different contexts, including home, medical settings, school, and everyday life. These emotions are *joy, sadness, anger, fear, surprise, disgust, frustration, embarrassment, boredom, nervousness, melancholy, guilt, hope, and stress*. In conclusion, from our standpoint, an ideal emotional dataset should demonstrate a well-balanced representation of data from various perspectives. To accomplish this objective, we conducted additional analysis on the dialogues generated using ChatGPT. This involved calculating several metrics, including the distribution of gender, the types of contexts, the consistency between emotions and contexts, and, overall, an evaluation of the interaction quality. By following this main trend, it is worth to notice that, recently, there has been a trendy interest on the issue of coupling HRI with *big data*, as it emerges from recent studies (e.g., [29, 10, 56]).

The remainder of the paper is organized as follows. The next Section explores some works on topics related to the theme of our study; in Section 3 we illustrate the methodology that we used to build the dataset; in Section 4 we discuss about results, and a sample of the collected and modified dialogues as well as the subsequent analysis is reported; then in Section 5 we present a case of study; subsequently in Section 6 a brief discussion is given about the dataset characteristics; in the end conclusions and future work are illustrated.

2. Related Work

There are several contributions in the literature in the area of developing models for emotion recognition in conversational agents. An interesting review was conducted by [46]. The study provides a systematic review of approaches in building an emotionally-aware chatbot (EAC). In [55] the authors use multi-task learning to predict the emotion label and generate a valid response for a given utterance. Their model consists of a self-attention based encoder and a decoder with dot product attention mechanism to generate response with a specified emotion and produce more emotionally relevant responses. In order to improve user interaction in [23] a model was developed with the task of generating empathetic and personalized dialogues, giving the machine the ability to respond emotionally and in accordance with the user. Other studies have tried to create empathetic generative chatbots that could simulate the responses provided by mental health professionals, mapping the emotions of the interlocutor and generating appropriate responses [27], [14], [20]. In general, a lot of efforts have been made to try to make conversational models more fluid by providing an ontology for the description of all concepts that may be relevant conversational topics, as well as their relationships to each other [26] or providing context and increasing the domain of knowledge [51], [59], [45], [39], [28]. Other interesting contributions train agents using datasets that include everyday conversations in order to dynamically answer people's questions. In [4] the chatbot model is based on the Simple Dialog and Daily Dialog datasets, which are then merged into a single dataset. Also, in [61] the authors train a model on 147 million conversational exchanges extracted from Reddit comment chains over a period from 2005 to 2017. In [49] the authors propose a new benchmark for generating empathetic dialogues and a new dataset based on emotional situations (EmpatheticDialogues) that includes 25k conversations.

Balanced dataset construction is critical for emotion recognition in general, but in [60] the authors point out that this is even more important when dealing with textual datasets. The reason is twofold: on the one hand because of the manual annotation of the data, a costly and time-consuming process, on the other hand because of the assignment of labels, which most of the time is related to subjectivity. In [62] they propose a Knowledge Enriched Transformer (KET), where contextual utterances are interpreted using hierarchical self-attention and external commonsense knowledge is dynamically leveraged using a context-aware affective graph attention mechanism At the same time, another crucial element is the techniques used for emotion recognition. In [31] the authors show the classification of emotion detection methods proposed in the literature: Keyword-based detection, Learning-based detection and Hybrid detection. In Keyword-based detection, emotions are detected based on the related set of keywords found in the input text. In Learning-based detection method, emotions are de-

tected based on previous training result. In Hybrid detection method, emotions are detected on the basis of a combination of detected key words, learned patterns, and other additional information. In [21] authors reviewed some deep learning-based technologies commonly utilized in Textual Emotion Recognition (TER). The most commonly used are: Word Embedding, Basal Neural Networks and Derived Variations, Knowledge Enhanced Representation and Transfer Learning for Emotion Recognition. Textual emotion in dialogue is dynamic and strongly correlated with context information, thus the TER of dialogue is more challenging. There are several methods that try to address this problem, as they reported in their survey. The first is context modeling, as each utterance is highly dependent on contextual clues, and good results will not be obtained if the sentence-level deep TER algorithm is directly applied to the dialogue. In particular, context modeling can be summarized as either a flatten context modeling or hierarchical context modeling. By flatten context modeling, context utterance and current utterance are concatenated, and all tokens are flattened into a word sequence. Then, a neural network receives this sequence of words for contextual learning and final prediction. However, the flattened context processing turns the word sequence too long and ignores the temporal step, disrupting the hierarchy. In hierarchical context modeling, each utterance in dialogue is embedded into a vector representation by utterance-level encoder: context information is further extracted by hierarchy context-level encoder. Another method is the Dynamic Emotion Modeling that mainly focus on tracking the contextual information and exploring the overall tone in dialogue.

Lastly, to the best of our knowledge, there are no other studies that have used ChatGPT to construct text datasets. ChatGPT is used in several areas, as highlighted in this interesting review [24]. The main industries that are exploiting the potential of ChatGPT are e-commerce companies, healthcare organizations, financial institutions, customer service, call centers, and potentially could be greatly exploited by education industry. Developers also benefit a lot from ChatGPT, e.g., for bug fixing, identifying potential errors in the system, and making new pieces of source code based on samples, but it seems that it has not been used to create text datasets.

3. The Proposed Methodology

Our work aims to construct a dataset in Italian for emotion recognition based on dialogues. Initially, we established methodological criteria to guide the generation process, and then we utilized ChatGPT to expedite the data generation phase. After generating the dialogues, professional psychologists reviewed each conversation to assess the dataset's suitability from various perspectives. Our analysis focused on examining the consistency between the intended emotion and the context, the distribution of genders, the types of contexts produced, and the quality of interaction. In this context, interaction quality refers to the appropriateness of language concerning specific emotions. The methodology employed in this study consists of three stages: the procedure for generating dialogues, data analysis, and subsequent improvements.

3.1. Main Procedure

We created twenty-five dialogues for each of the fourteen emotions in our dataset. The instruction given to ChatGPT was to generate a brief conversation, approximately five lines long, between two individuals, wherein a specific emotion is evident. Additionally, we generated five more dialogues for each emotion, requesting ChatGPT to avoid using the word associated with that emotion. These particular dialogues were labeled as "Without Word (W.W.)".

This was done to test whether ChatGPT could generate discussions in which, e.g., sadness emerged without having the word “sadness" in the text. The goal is to create data that increasingly reflect real situations to train robots that can recognize emotions based on context and not just by recognizing specific words. The small number is because this is a pilot study to build a more extensive dataset later. Finally, the original dialogues generated were retained, but we created a copy to edit them after performing the analysis. Both the Web interface and the API provided by OpenAI were used. This has made it possible to obtain different styles of narrations of the events. Gpt 3.5-turbo model was used, with the following role: “You are a writer assistant who produces dialogue that accurately reflects emotion".

3.2. Analysis

Dialogues were analyzed considering four factors: consistency between context and emotion, gender distribution, type of contexts, and quality of interaction. By **consistency (C)** between context and emotion, we mean whether the context generated is consistent with the feeling expressed. For example, the context of an argument with the boss is a context compatible with the emotion of anger. So for each dialogue, we assessed whether or not there was consistency. We counted the percent relative frequency.

$$C = \frac{Nyes}{Ndialogues} \cdot 100$$

Similarly, for **gender distribution(GD)**, we counted how many times the gender “Neutral (N), Masculine (M) and Feminine (F)" occurred in the dialogues and we calculated the percent relative frequency.

$$GD = \frac{Ngender(NorMorF)}{Ntotgender} \cdot 100$$

Regarding the **type of context(TC)**, we created classes and counted how many belonged to each class; then, we calculated the percent relative frequency.

$$TC = \frac{NcontextX}{Ntotcontexts} \cdot 100$$

The classes identified are *Work*, *Leisure*, *Luck*, *Interpersonal sphere*, *Generic*. In some cases, we identified a specific category, e.g., in the “Disgust" dialogues, we identified

the category “Animals and Objects," as several scenarios expressed disgust for objects or animals.

Finally, for the **quality of interaction(QoI)**, we analyzed the appropriateness of language in expressing a specific emotion. This was evaluated with three values: “*Sufficient*”, “*Not much*”, “*No*”. By “Sufficient (S)” we mean that the language appears natural enough and reflects in the terms used the emotion. By "Not much (NM)" we mean that the language is not very natural and it does not entirely reflect the emotion, e.g., using words that also represent other emotions, but all in all, it is acceptable. By "No (N)," we mean confusion, unusual terms, and/or language that does not reflect the specific emotion. Also, for this parameter, we calculated the percent relative frequency.

$$QoI = \frac{NValue(SorNMorN)}{Ntotinteractions} \cdot 100$$

3.3. Further Improvements

Following the analysis, we implemented several modifications addressing grammatical and content-related aspects. An important focus was placed on examining the distribution of different contexts and selecting those most relevant to interpersonal and social scenarios for inclusion in the dataset, which will be built after this pilot study.

To ensure diverse scenarios, we initially requested the generation of five potential social scenarios where a specific emotion could manifest. This approach enabled us to identify scenarios aligned more closely with Human-Robot Interaction (HRI). Once an interesting scenario was generated, we requested modifications to emphasize social interaction. A dialogue was created and expanded for each of these scenarios as needed. However, the model often struggled to broaden the conversation without repetitively using emotion-specific terms. To address this, we requested the model to replace such terms with expressions that could serve as metaphors or equivalent phrases.

When changing scenarios, there were instances where certain emotions were confused. For example, when requesting the feeling of anger, the generated dialogues often expressed frustration, frequently using the term "frustrating" in the text, and vice versa. Similar confusion occurred with stress and nervousness. To overcome this, we initially asked for a clear definition to differentiate between the two emotions. Once provided with the description, we instructed the model to generate scenarios where each emotion could emerge distinctly. As a result, the developed scenarios became more specific, effectively distinguishing between the two emotions. The same approach was applied to stress and nervousness. This process underscores the significance of human experts intervening in all phases to guide ChatGPT in generating more focused dialogues.

4. Empirical Results

The results will be shown first according to a global view and then in detail for each emotion.

4.1. Global Analysis

With respect to **consistency**, 86% of the generated contexts are consistent with emotion. An example of consistency is this:

- Person 1: Ciao, come stai oggi? (*Hello, how are you today?*)
- Person 2: Non molto bene, sinceramente. (*Not very well, honestly.*)
- Person 1: Oh, mi dispiace sentirlo. C’è qualcosa che ti preoc*cupa? (Oh, I’m sorry to hear that. Is something bothering you?*)
- Person 2: Sì, ho avuto un brutto litigio con un caro amico ieri sera. Siamo finiti per dire cose terribili l’un l’altro e ora mi sento come se avessi perso una parte di me stesso. (*Yes, I had a bad fight with a close friend last night. We ended up saying terrible things to each other and now I I feel like I’ve lost a part of myself.*)
- Person 1: Capisco. Litigi del genere sono sempre molto difficili da gestire. Ma forse potresti cercare di parlare con il tuo amico e chiarire la situazione? (*I understand. Fights like that are always very difficult to deal with. But maybe you could try to talk to your friend and clear the air?*)
- Person 2: Ho provato a chiamarlo stamattina, ma non ha risposto. Non so se vuole ancora parlarmi. (*I tried to call him this morning, but he didn’t answer. I don’t know if he still wants to talk to me.*)
- Person 1: Mi dispiace tanto. Spero che alla fine riuscirete a risolvere le cose. Se hai bisogno di parlare, sai che puoi contare su di me. (*I’m so sorry. I hope you will eventually be able to work things out. If you need to talk, you know you can count on me.*)
- Person 2: Grazie, apprezzo molto il tuo sostegno. Ma per ora mi sento solo triste e confuso. (*Thank you, I really appreciate your support. But for now I just feeling sad and confused.*)

In this dialogue, the context is an argument with a friend, and effectively the sadness that emerges is consistent with the type of context. In some cases, the generated contexts did not reflect the corresponding emotion. An example:

- Person 1: Ciao, come stai? (*Hello, how are you?*)
- Person 2: Non molto bene, devo dire. (*Not very well, I must say.*)
- Person 1: Che cosa è successo? (*What happened?*)
- Person 2: Stamattina mi sono svegliato con un forte mal di stomaco. Ho passato tutto il mattino in bagno. (*I woke up this morning with a bad stomachache. I spent the whole morning in the bathroom.)*

- Person 1: Ah, mi dispiace sentirlo. Cosa pensi possa essere? (*Ah, I'm sorry to hear that. What do you think it might be?)*
- Person 2: Non ne sono sicuro, ma credo sia stato qualcosa che ho mangiato ieri sera. Ho provato un nuovo ristorante messicano, ma credo che il cibo non fosse fresco. (*I'm not sure, but I think it was something I ate last night. I tried a new Mexican restaurant, but I think the food wasn't fresh.*)
- Person 1: Oh no, che schifo. Ti senti meglio adesso? (*Oh no, that's gross. Do you feel better now?*)
- Person 2: Un po' meglio, ma non ancora del tutto. Devo ancora prendere qualche medicina. (*A little better, but not quite yet. I still have to take some medicine.*)
- Person 1: Sì, è meglio prendersi cura di te stesso. Spero tu ti riprenda presto. (*Yes, you better take care of yourself. I hope you recover soon.*)

In this case, except for "that's gross", the entire dialogue is centered on a context showing a reaction to physical discomfort. The person feels vomiting because of physical discomfort, as he has probably eaten bad food, and therefore not because of an emotional reaction of disgust.

Regarding **gender**, there is a strong imbalance in the dataset: N=30% M=56% F=14%

In addition, in a couple of cases, the gender count was canceled because the same person was first male and then female. Here is an example of a dialogue about frustration:

- Person 1: Ho lavorato duramente su questo progetto ma non ha (grammar error) ottenuto il successo sperato. (*I worked hard on this project but it did not (in the Italian version-grammar error) achieve the success I had hoped for.*)
- Person 2: Mi dispiace sentirti così deluso (indicates that person 1 is male). Cosa pensi sia andato storto? (*I'm sorry to feel so disappointed (in the Italian version indicates that person 1 is male). What do you think went wrong?*)
- Person 1: Non ne sono sicuro, ho messo tutta me stessa (female gender) ma sembra che non sia abbastanza. (*I'm not sure, I put all of myself (in the Italian version-female gender) but it seems like it's not enough.*)
- Person 2: Non scoraggiarti, ogni esperienza è una lezione imparata. Magari hai bisogno di un po' di tempo per riflettere e riprovarci con un approccio diverso. (*Don't be discouraged, every experience is a lesson learned. Maybe you need some time to reflect and try again with a different approach.*)

The **context** overall appears heterogeneous but it is unbalanced when observed in relation to specific emotions. For example, for the emotion "Joy," only three types of context were generated. Specifically, ten contexts are about *success* (e.g., passing a university exam, promotion at work), ten are about *leisure* (e.g., traveling, starting a yoga class), four are about *luck* (e.g., winning the lottery), and only one is about *Personl life situations* (receiving a gift). The type of context will be discussed in depth in the description of each emotion.

Regarding the **quality of interaction**, the adherence of text to the desired manifestation of emotions was evaluated. In 65% of the dialogues, we can define the quality of interaction as "sufficient". However, some changes were added later either in terms of grammatical corrections or to make the dialogue more fluid and natural. In 25% of cases, there is a poor fit between text and emotion. Finally, in 10% of the dialogues, the text was completely garbled or did not reflect the desired emotion. Here are some examples of the three categories:

Sufficient: Boredom

- Friend 1: "Cosa c'è che non va, sembri distratta?" (*"What's wrong, you seem distracted?"*)
- Friend 2: "Sì, sto solo pensando ad altro. Questa lezione mi fa venire la noia." (*"Yes, I'm just thinking about something else. This class is making me bored."*)
- Friend 1: "Capisco come ti senti, anche io sto trovando difficoltà a restare concentrata." (*"I understand how you feel, I am also finding it hard to stay focused."*)
- Friend 2: "Sì, vorrei solo essere altrove ora. Anche voi pensate la stessa cosa, giusto?" (*"Yes, I just wish I was somewhere else right now. You also feel the same way, right?"*)
- Friend 1: "Sì, credo che tutti vorremmo fare altro." (*"Yes, I think we all would like to do something else."*)

Not much: Anger

- Person 1: "Non ci posso credere, ho perso l'autobus per la terza volta questa settimana!" (*"I can't believe I missed the bus for the third time this week!"*)
- Person 2: "Ma come hai fatto?" (*"But how did you do it?"*)
- Person 1: "Non mi hai visto? Mi hai tenuto a parlare e l'autobus è passato sotto il mio naso!" (*"Didn't you see me? You kept me talking and the bus passed right under my nose!"*)
- Person 2: "Non è colpa mia se sei sempre in ritardo!" (*"It's not my fault you're always late!"*)
- Person 1: "Ma certo che è colpa tua! Non riesci mai a smettere di parlare e poi ti lamenti se arrivo sempre tardi!" (*"Of course it's your fault! You can never stop talking and then you complain that I'm always late!"*)

- Person 2: "Ok, ok, calmati! Non c'è bisogno di arrabbiarsi!" (*"Okay, okay, calm down! No need to get angry!"*)
- Person 1: "Ma come faccio a non arrabbiarmi? Questo mi fa perdere tempo e soldi!" (*"But how can I not get angry? This wastes my time and money!"*)
- Person 2: "Hai ragione, mi dispiace. Cercherò di essere più attento la prossima volta." (*"You're right, I'm sorry. I'll try to be more careful next time."*)

No: Hope

- Character A: "Spero solo di non sembrare troppo stressato/a stasera." (*"I just hope I don't look too stressed out tonight."*)
- Character B: "Non preoccuparti, sei bellissimo/a e la serata sarà fantastica." (*"Don't worry, you look beautiful and the evening will be great."*)
- Character A: "Speriamo che ci siano delle sorprese piacevoli stasera, vorrei che fosse tutto diverso dal solito." (*"Hopefully there will be some pleasant surprises tonight, I'd like everything to be different than usual."*)
- Character B: "Stasera sarà diversa dal solito, perché sarà proprio come ci piace. Semplice e piena di speranze!" (*"Tonight will be different than usual, because it will be just the way we like it. Simple and hopeful!"*)

4.2. Single Emotion Analysis

Below we show the analysis of each of the 14 emotions according to the 4 parameters outlined in the methodology section.

- **JOY**
  - Consistency = 100%
  - Gender = N 12% M 80% F 8%
  - Contexts = 10 Success, 10 Leisure, 4 Luck, and only 1 is about personal life situations
  - Quality of interaction = Sufficient 64% Not much 36%
- **SADNESS**
  - Consistency = 92%
  - Gender = N 46% M 54% F 0
  - Contexts = Heterogeneous mainly generic and interpersonal
  - Quality of interaction = Sufficient 88% Not much 12%
- **ANGER**
  - Consistency = 100%
  - Gender = N 12% M 55% F 33%
  - Contexts = Heterogeneous, sometimes reactions out of proportion to the context
  - Quality of interaction = Sufficient 88% Not much 12%
- **FEAR**
  - Consistency = 100%
  - Gender = N 24% M 72% F 4%
  - Contexts = Mostly related to horror contexts (shadows, animals, running away from someone) - Absence of contexts related to more interpersonal or social fear, such as fear of the future.
  - Quality of interaction = Satisfactory 72% Not much 28%
- **SURPRISE**
  - Consistency = 100%
  - Gender = N 24% M 76% F 0%
  - Contexts = Heterogeneous
  - Quality of interaction = Satisfactory 88% Not much 12%
- **DISGUST**
  - Consistency = 96%
  - Gender = N 72% M 28% F0
  - Contexts = Highly related to foods, insects, objects. No examples related to people's behaviors or abstract concepts. Only in two cases is there a reference to disgust as a result of a person's behavior.
  - Quality of interaction = Sufficient 84% Not much 16%
- **FRUSTRATION**
  - Consistency = 28%: in three cases there is confusion with *anger*
  - Gender = N 46% M 50% F 4%
  - Contexts = Heterogeneous, sometimes reactions out of proportion to the context
  - Quality of interaction = Sufficient 80% Not much 20%
- **EMBARRASSMENT**
  - Consistency = 68% sometimes there is confusion with *guilt*.
  - Gender = N 44% M 48% F 8%
  - Contexts = Heterogeneous
  - Quality of interaction = Sufficient 76% Not much 24%

- **BOREDOM**
  - Consistency = 92%
  - Gender = N 16% M 56% F 28%
  - Contexts = Heterogeneous, mainly leisure time
  - Quality of interaction = Sufficient 56% Not much 28% No 16%
- **NERVOUSNESS**
  - Consistency = 88%
  - Gender = N0 M 53% F 47%
  - Contexts = Heterogeneous
  - Quality of interaction = Sufficient 68% Not much 20% No 12%
- **MELANCHOLY**
  - Consistency = 88%
  - Gender = N 40% M 60% F0
  - Contexts = Heterogeneous
  - Quality of interaction = Sufficient 68% Not much 16% No 16%
- **GUILT**
  - Consistency = 92%
  - Gender = N 40% M 40% F 20%
  - Contexts = 24% relate to work contexts, while most are related to interpersonal or social situations (e.g., arguing with a friend, neglecting family, telling a lie, etc...)
  - Quality of interaction = Satisfactory 52% Not much 44% No 4% . In many dialogues the language appears out of proportion to the emotion
- **HOPE**
  - Consistency = 100%
  - Gender = N 52% M 36% F 16%
  - Contexts = 28% relate to work contexts, 44% relate to medical contexts, 28% relate to interpersonal or social situations
  - Quality of interaction = Satisfactory 28% Not much 64% No 8% . Often the language seems to belong more to fear or nervousness and not to hope. Here is an example:

- Student 1: "Sto preparando questo esame da giorni, spero di ottenere un buon voto." (*"I've been preparing for this exam for days, I hope to get a good grade."*)
- Student 2: "Sono sicuro che andrà tutto bene, hai studiato tanto e sai quello che fai." (*"I'm sure you'll do well, you've studied hard and you know what you're doing."*)
- Student 1: "Sì, ma ho paura di non ricordare tutte le informazioni durante l'esame." (*"Yes, but I'm afraid I won't remember all the information during the exam."*)
- Student 2: "Non preoccuparti, vai tranquillo e non lasciare che l'ansia ti prenda il sopravvento. Spero che otterrai la valutazione che meriti." (*"Don't worry, go easy and don't let anxiety get the best of you. I hope you will get the grade you deserve."*)
- Student 1: "Grazie per il supporto! Ho davvero bisogno di sentirlo. Speriamo che andrà tutto bene." (*"Thank you for the support! I really need to hear it. Hopefully everything will be okay."*)
- Student 2: "Sarà così, cerca di rilassarti e di essere positivo. La speranza è la chiave del successo." (*"It will, just try to relax and be positive. Hope is the key to success."*)

- **STRESS**
  - Consistency = 84% . Sometimes confusion with nervousness
  - Gender = N 0 M 83% F 17%
  - Contexts = 56% relate to work contexts, the remaining are heterogeneous
  - Quality of interaction = Satisfactory 60% Not much 12% No 28%

4.3. Analysis of Dialogues "Without Words"

As for dialogues generated without the word expressing the emotion emerging, most generated contexts **consistent** with the emotion. Not very consistent were the dialogues of Boredom, Fear, Surprise and Hope. In contrast, only 1 out of 5 of the Disgust dialogues was found to be consistent. The other 4 were so confused that they could not be classified. Here is an example:

- Person 1: "Che schifo...quella lì sembra venuta direttamente dal nido delle rane!" (*"Gross...that one looks like it came straight from the frogs' nest!"*)
- Person 2: "Davvero, mi ha fatto venire il voltastomaco. Però non possiamo farci niente al riguardo... forse conviene che andiamo a prendere un po' d'aria fresca." (*"Really, it made me sick to my stomach. We can't do anything about it though...maybe we'd better go get some fresh air."*)
- Person 1: "Sì, direi che hai ragione. Quando si vedono situazioni simili, l'unica cosa da fare è portare il nonno in braccio in giro per la città, almeno la compagnia è più gradevole!" (*"Yes, I would say you are right. When you see situations like that, the only thing to do is to carry Grandpa around town, at least the company is more pleasant!"*)

Furthermore, right among the dialogues of Disgust-W.W. a stereotype about Neapolitans emerged. Here is the dialogue:

- Person 1: "Oh, guardate quella lì...è più volgare di un bidone di spazzatura a Napoli!" (*"Oh, look at that one...she's grosser than a garbage can in Naples!"*)
- Person 2: "Sì, mi ha messo i brividi appena l'ho vista. Comunque, perché non ci beviamo un po' d'acqua insieme? Così ci togliamo lo schifo di bocca!" (*"Yeah, she gave me the creeps as soon as I saw her. Anyway, why don't we have some water together? That way we can get the filth out of our mouths!"*)
- Person 1: "Mi pare un'ottima idea, non vedo l'ora di liberarmi di questa sensazione." (*"That sounds like a great idea, I can't wait to get rid of this feeling."*)

It is not only not at all sufficient from the point of view of language, but a stereotype clearly emerges. Regarding **gender** and **contexts**, the number of dialogues is small to draw specific inferences, however, we can say that they seem to reflect the general trend. As for the **quality of interaction**, it appears worse than the basic dialogues, that is, the non-W.W. dialogues. In fact, in 43% of the cases the quality of interaction was rated as "sufficient", in 32% of the cases "not very much", and in 25% "no". The sum of "not very much" and "no" is also 57% thus exceeding the percentage of those considered sufficient.

5. Case Study

The dialogues created with ChatGPT were used as dialogue prototypes to improve human-robot interaction. Specifically, we used the RiveScript [3] dialogue engine to create a simple system that could give the ability to create a natural language interaction system associated with specific situations. RiveScript employs a concise set of rules that, when combined, can provide effective chatbot personalities. By writing triggers in a simplified regular expression format, complex sets of word patterns can be efficiently matched in a single step, enhancing the chatbot's capabilities. The core library is compact, self-contained, and it is aimed at receiving human input and delivering "intelligent", even if pattern-matching based, responses. This adaptability allows RiveScript to be utilized according to individual needs, empowering users with flexibility. One of the main advantages of the framework is that itadopts a straightforward plain text scripting language that is easy to learn and allows quick writing. Its line-based structure is readily understandable for maintenance purposes without using cumbersome XML code or complex symbols, which may hide the code structures.

Starting with the stimulus-response pair, which is the basic element of the RiveScript knowledge base, ChatGPT was asked to generate a set of N sentences similar to the input (trigger) sentences and N sentences semantically similar to the output answers of the bot. The phrases generated as triggers are provided into a subsymbolic layer that is used to identify what the user is saying. To build such a sub-symbolic layer, we have exploited the Spacy library [2], which easily allows to compute a sub-symbolic semantic similarity between two sentences. At the same time, the response phrases are used to enhance the variety of chatbot replies to make the interaction more engaging. To be more specific, we formulate the same dialogue (context, type of emotion, type of interaction) while increasing the variety of possible words (as it might happen in real life). We thus exploit "fast data generation" to improve the robot's listening skills and avoid monotonicity in responses.

An advantage is that we can exploit the dataset for emotion recognition as an additional element for emotion detection according to a given context. Specifically, the system is coded as a finite state machine that identifies the sequence of sentences uttered by the user based on the sentences given by the chatbot and contextualizes the situation, providing an extra element to associate the contextual interaction with a possible emotion, which has somehow been implicitly associated with the dialogue. In this manner, the agent will have a "basic" dialogue skeleton and a set of possible variations, all of them considering a specific context and emotion related to the interaction. This also improves the quality of the interaction, the associative process regarding emotions and context, and the possibility of understanding the situation.

As mentioned above, for each sentence in each chatgpt-generated dialogue, chatGpt was then asked to provide a number N of similar sentences. These sentences play a similar role to the emotional beacons introduced in [47]. In particular, given $s_{i,d}$ the *i-th* sentence of the *d-th* dialogue, a set of N sentences $\{b^j_{i,d}\}$ is generated with $j = 1, \cdots, N$ where N is a positive integer that has been set to $N = 10$ for our case study. Each sentence $\{b^j_{i,d}\}$ together with each "seed" sentence $s_{i,d}$ is then encoded subsymbolically. For efficiency reasons, we used the Spacy library [2] and the Italian language model "it_core_news_sm". Each sentence u_k provided to the system by the user is then compared sub symbolically with one of the possible sentences in the chatgpt-generated dialogues $s_{i,d}$ and the related "beacons" $\{b^j_{i,d}\}$ which are expected to be said by the user: the semantic similarity between the sentence u_k and all sentences $s_{i,d}$ and $\{b^j_{i,d}\}$ is then computed. If the calculated similarity exceeds a certain experimentally established threshold T_h, the corresponding dialogue d with which the sentence can be associated is identified. If several sentences belonging to different dialogues provide semantic similarity greater than the threshold T_h, the dialogue with the highest similarity value is selected.

Once a specific dialogue has been identified, the system tries to maintain the same structure provided by chatGPT to identify a possible context that is already known. The comparison then between what the user says and the sentences, along with their beacons, that the system expects continues until the presumed conclusion of the dialogue and recognition of the situation. If, during the conversation, the calculation of semantic similarity within the selected dia-

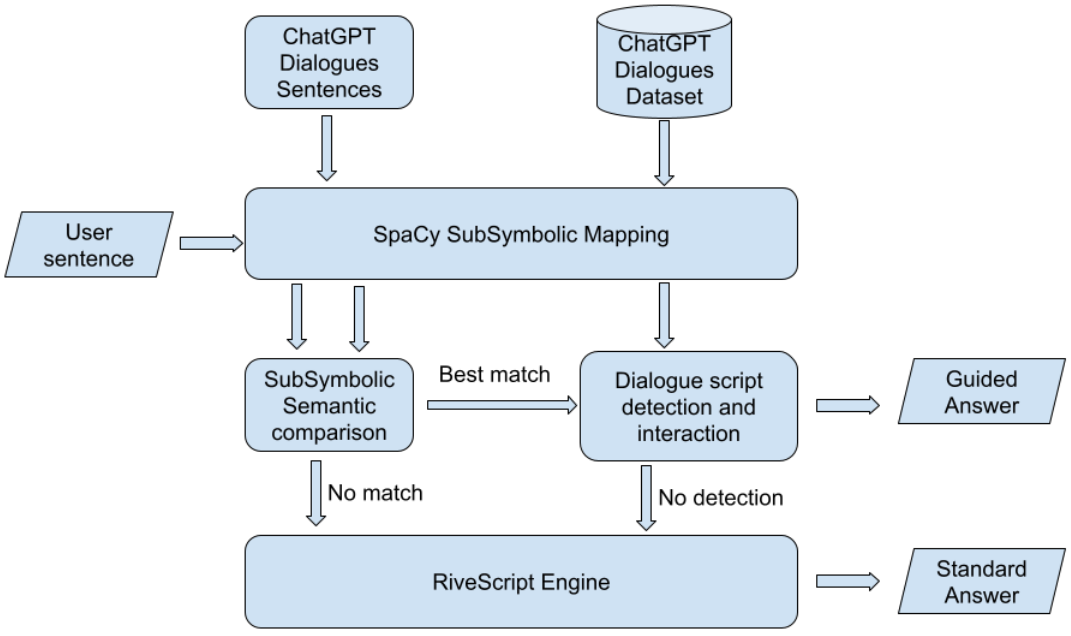


Figure 1: A schema of the interaction process for the case of study.

logue gives a value below the minimum value required by the threshold T_h, the system restarts from scratch, trying to identify another possible discussion. If, in the end, the sub-symbolic comparison of semantic similarity does not yield any value above the predetermined T_h threshold value, the control of the interaction passes directly to the RiveScript engine, which continues with the chat according to the usual standard rule-based chatbot mode. In any case, each sentence spoken by the user is still sub symbolically encoded and compared with the extended set of sentences in the dataset of known dialogues to identify possible occurrences of proper contexts to be recognized for determining the emotions expressed by the user during the interaction with the robot. A schema of the adopted architecture for this case of study is reported in Fig. 1.

6. Critical Discussion

The analysis of the dataset revealed both strengths and weaknesses, providing valuable insights for constructing the larger dataset. The main strength lies in the consistency between context and emotion, which ensures high reliability in automatically generating dialogues and facilitates fast data generation. However, as the results indicate, the generated conversations need to undergo validation by a human operator, as they were not consistently consistent. Additionally, special attention must be given to the types of contexts included in the dataset, aiming for maximum heterogeneity. This can be achieved by incorporating increasingly relevant contexts to interpersonal and social spheres, thereby reflecting realistic Human-Robot Interaction (HRI) scenarios.

The analysis of gender distribution highlighted a significant imbalance favoring male gender representation. This finding allows for addressing the bias and encourages broader reflections on training AI systems that strive for greater diversity. Furthermore, the presence of stereotypes concerning the city of Naples in some dialogues should also be carefully considered.

Lastly, the quality of interaction frequently necessitates modifications by the human operator. This may involve rectifying occasional grammatical errors, aligning the language with the intended emotion, and enhancing the naturalness of the dialogues.

Regarding the case study, we believe that a system designed in this way can be of help in improving HRI by having some specific strengths. One of these is the presence of context in dialogues: this can help the robot place certain emotions in certain contexts making the interaction more consistent. Another element is the variety of words expressing the same concept: this allows the robot to respond in a relevant way even when there are word variations as there are in natural interactions. Clearly, the effectiveness of the model will be the subject of subsequent studies, which will aim to identify measurement tools to establish the actual improvement of HRI.

7. Conclusions and Future Work

We conducted an initial study to guide the development of an Italian dataset designed for emotion recognition. Once the methodology and procedure were established, we leveraged ChatGPT to generate dialogues rapidly. Working alongside psychology experts, we meticulously examined 420 conversations covering 14 emotions to assess the dataset's balance across various dimensions, including consistency between context and emotion, gender distribution, types of generated context, and interaction quality. The find-

ings revealed the advantages and limitations of utilizing automated dialogue generation systems. It became evident that the construction of the dataset cannot disregard human control.

The most significant advantage observed was the speed of data generation, with most cases demonstrating consistency between the generated contexts and intended emotions. However, exercising control over the dialogues remains crucial to ensure heterogeneity and a stronger focus on interpersonal and social aspects. The study also highlighted a notable gender distribution imbalance, predominantly favoring masculine representations. Addressing this disparity will generate dialogues explicitly requesting feminine and neutral genders, thereby achieving a more balanced dataset.

Furthermore, numerous modifications were made to the dialogues concerning language usage and interaction quality, encompassing grammatical, structural, and content corrections. Despite these adjustments, an additional advantage emerged—the ability to create dialogues from scratch, directing ChatGPT to generate dialogues that align with predetermined criteria established by the researchers. Moving forward, this study's insights will inform the development of a larger, well-balanced dataset tailored specifically for (HRI) scenarios.

Finally, the case study allowed us to observe how fast data generation is, making it possible to create more examples, thus improving the robot's listening skills and avoiding the monotonicity of responses. Certainly, however, a deeper evaluation of effectiveness will be the subject of subsequent studies.On the other hand, we aim at integrating our framework with emerging challenges due to novel *big data trends* (e.g., [8, 18, 5, 38, 17, 35, 36, 37, 6, 22, 34]).

Acknowledgements

The authors would thank Mr. Antonino Asta for his partial contribution to this work.

References

[1] , . Crowdflower. 2016. the emotion in text. https://www.figure-eight.com/data/ sentiment-analysis-emotion-text/.

[2] , . Industrial-strength natural language processing. https://spacy.io/.

[3] , . Rivescript. https://www.rivescript.com/.

[4] Bachtiar, F.A., Fauzulhaq, A.D., Manullang, M.T.R., Pontoh, F.R., Nugroho, K.S., Yudistira, N., 2022. A generative-based chatbot for daily conversation: A preliminary study, in: 7th International Conference on Sustainable Information Engineering and Technology 2022, pp. 8–12.

[5] Balbin, P.P.F., Barker, J.C.R., Leung, C.K., Tran, M., Wall, R.P., Cuzzocrea, A., 2020. Predictive analytics on open big data for supporting smart transportation services. Procedia Computer Science 176, 3009–3018.

[6] Barkwell, K.E., Cuzzocrea, A., Leung, C.K., Ocran, A.A., Sanderson, J.M., Stewart, J.A., Wodi, B.H., 2018. Big data visualisation and visual analytics for music data mining, in: 22nd International Conference Information Visualisation, IV 2018, Fisciano, Italy, July 10-13, 2018, IEEE Computer Society. pp. 235–240.

[7] Basile, V., Maria, D.M., Danilo, C., Passaro, L.C., et al., 2020. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and sspeech tools for italian, in: Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), CEUR-ws. pp. 1–7.

[8] Bellatreche, L., Cuzzocrea, A., Benkrid, S., 2010. *F&A*: A methodology for effectively and efficiently designing parallel relational data warehouses on heterogenous database clusters, in: Data Warehousing and Knowledge Discovery, 12th International Conference, DAWAK 2010, Bilbao, Spain, August/September 2010. Proceedings, Springer. pp. 89–104.

[9] Braunhofer, M., Elahi, M., Ricci, F., 2015. User Personality and the New User Problem in a Context-Aware Point of Interest Recommender System. pp. 537–549. doi:10.13140/RG.2.1.2493.8001.

[10] Budiharto, W., Andreas, V., Gunawan, A.A.S., 2020. Deep learning-based question answering system for intelligent humanoid robot. J. Big Data 7, 1–10.

[11] Buechel, S., Hahn, U., 2017. EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain. pp. 578–585. URL: https://aclanthology.org/E17-2092.

[12] Caruccio, L., Cirillo, S., Deufemia, V., Polese, G., Stanzione, R., 2022. Data analytics on twitter for evaluating women inclusion and safety in modern society, in: Proceedings of the 1st Italian Conference on Big Data and Data Science (ITADATA 2022), Milan, Italy, September 20-21, 2022, CEUR Workshop Proceedings.

[13] Caselli, T., Novielli, N., Patti, V., Rosso, P., 2018. Sixth evaluation campaign of natural language processing and speech tools for italian: Final workshop (evalita 2018), in: EVALITA 2018, CEUR Workshop Proceedings (CEUR-WS. org).

[14] Catania, F., Spitale, M., Fisicaro, D., Garzotto, F., 2019. Cork: A conversational agent framework exploiting both rational and emotional intelligence., in: IUI Workshops.

[15] Chiavetta, F., Bosco, G.L., Pilato, G., et al., 2016. A lexicon-based approach for sentiment classification of amazon books reviews in italian language. WEBIST (2) 2016, 159–170.

[16] Corazza, M., Menini, S., Cabrio, E., Tonelli, S., Villata, S., 2020. A multilingual evaluation for online hate speech detection. ACM Transactions on Internet Technology (TOIT) 20, 1–22.

[17] Coronato, A., Cuzzocrea, A., 2022. An innovative risk assessment methodology for medical information systems. IEEE Trans. Knowl. Data Eng. 34, 3095–3110.

[18] Cuzzocrea, A., Martinelli, F., Mercaldo, F., Vercelli, G.V., 2017. Tor traffic analysis and detection via machine learning techniques, in: IEEE BigData, 2017, IEEE Computer Society. pp. 4474–4480.

[19] Cuzzocrea, A., Pilato, G., 2021. A composite framework for supporting user emotion detection based on intelligent taxonomy handling. Logic Journal of the IGPL 29, 207–219.

[20] Das, A., Selek, S., Warner, A.R., Zuo, X., Hu, Y., Keloth, V.K., Li, J., Zheng, W.J., Xu, H., 2022. Conversational bots for psychotherapy: A study of generative transformer models using domain-specific dialogues, in: Proceedings of the 21st Workshop on Biomedical Language Processing, pp. 285–297.

[21] Deng, J., Ren, F., 2021. A survey of textual emotion recognition and its challenges. IEEE Transactions on Affective Computing .

[22] Filali, H., Riffi, J., Boulealam, C., Mahraz, M.A., Tairi, H., 2022. Multimodal emotional classification based on meaningful learning. Big Data Cogn. Comput. 6, 95.

[23] Firdaus, M., Thangavelu, N., Ekbal, A., Bhattacharyya, P., 2022. I enjoy writing and playing, do you: A personalized and emotion grounded dialogue agent using generative adversarial network. IEEE Transactions on Affective Computing .

[24] George, A.S., George, A.H., 2023. A review of chatgpt ai's impact on several business sectors. Partners Universal International Innovation Journal 1, 9–23.

[25] Goel, A., Gautam, J., Kumar, S., 2016. Real time sentiment analysis of tweets using naive bayes, in: 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), IEEE. pp. 257–

261.

[26] Grassi, L., Recchiuto, C.T., Sgorbissa, A., 2022. Knowledge-grounded dialogue flow management for social robots and conversational agents. International Journal of Social Robotics 14, 1273–1293.

[27] Huang, J.Y., Lee, W.P., Chen, C.C., Dong, B.W., 2020a. Developing emotion-aware human–robot dialogues for domain-specific and goal-oriented tasks. Robotics 9, 31.

[28] Huang, M., Zhu, X., Gao, J., 2020b. Challenges in building intelligent open-domain dialog systems. ACM Transactions on Information Systems (TOIS) 38, 1–32.

[29] Jiang, M., Zhang, L., 2015. Big data analytics as a service for affective humanoid service robots, in: INNS Conference on Big Data 2015, San Francisco, CA, USA, 8-10 August 2015, Elsevier. pp. 141–148.

[30] Joshi, A., Tripathi, V., Bhattacharyya, P., Carman, M.J., 2016. Harnessing sequence labeling for sarcasm detection in dialogue from TV series 'Friends', in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany. pp. 146–155. URL: https://aclanthology.org/K16-1015, doi:10.18653/v1/K16-1015.

[31] Kao, E.C.C., Liu, C.C., Yang, T.H., Hsieh, C.T., Soo, V.W., 2009. Towards text-based emotion detection a survey and possible improvements, in: 2009 International conference on information management and engineering, IEEE. pp. 70–74.

[32] Kaur, R., Majumdar, A., Sharma, P., Tiple, B., 2023. Analysis of tweets with emoticons for sentiment detection using classification techniques, in: Distributed Computing and Intelligent Technology - 19th International Conference, ICDCIT 2023, Bhubaneswar, India, January 18-22, 2023, Proceedings, Springer. pp. 208–223.

[33] Koelstra, S., Muhl, C., Soleymani, M., Lee, J.S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., Patras, I., 2011. Deap: A database for emotion analysis; using physiological signals. IEEE transactions on affective computing 3, 18–31.

[34] Lee, S.J., Lim, J., Paas, L., Ahn, H.S., 2023. Transformer transfer learning emotion detection model: synchronizing socially agreed and self-reported emotions in big data. Neural Comput. Appl. 35, 10945–10956.

[35] Leung, C.K., Braun, P., Hoi, C.S.H., Souza, J., Cuzzocrea, A., 2019a. Urban analytics of big transportation data for supporting smart cities, in: Big Data Analytics and Knowledge Discovery - 21st International Conference, DaWaK 2019, Linz, Austria, August 26-29, 2019, Proceedings, Springer. pp. 24–33.

[36] Leung, C.K., Chen, Y., Hoi, C.S.H., Shang, S., Cuzzocrea, A., 2020a. Machine learning and OLAP on big COVID-19 data, in: 2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10-13, 2020, IEEE. pp. 5118–5127.

[37] Leung, C.K., Chen, Y., Hoi, C.S.H., Shang, S., Wen, Y., Cuzzocrea, A., 2020b. Big data visualization and visual analytics of COVID-19 data, in: 24th International Conference on Information Visualisation, IV 2020, Melbourne, Australia, September 7-11, 2020, IEEE. pp. 415–420.

[38] Leung, C.K., Cuzzocrea, A., Mai, J.J., Deng, D., Jiang, F., 2019b. Personalized deepinf: Enhanced social influence prediction with deep learning and transfer learning, in: IEEE BigData, 2019, IEEE. pp. 2871–2880.

[39] Li, Q., Li, P., Ren, Z., Ren, P., Chen, Z., 2022. Knowledge bridging for empathetic dialogue generation, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 10993–11001.

[40] Lim, M.J., Yi, M.H., Shin, J.H., 2023. Intrinsic emotion recognition considering the emotional association in dialogues. Electronics 12.

[41] Mohammad, S., Bravo-Marquez, F., 2017. WASSA-2017 shared task on emotion intensity, in: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics, Copenhagen, Denmark. pp. 34–49. URL: https://aclanthology.org/W17-5205, doi:10.18653/v1/W17-5205.

[42] Mohammad, S.M., 2017. Word affect intensities. arXiv preprint arXiv:1704.08798 .

[43] Mohammad, S.M., Bravo-Marquez, F., Salameh, M., Kiritchenko, S., 2018. Semeval-2018 Task 1: Affect in tweets, in: Proceedings of International Workshop on Semantic Evaluation (SemEval-2018), New Orleans, LA, USA.

[44] Mohammad, S.M., Sobhani, P., Kiritchenko, S., 2017. Stance and sentiment in tweets. ACM Transactions on Internet Technology (TOIT) 17, 1–23.

[45] Nuruzzaman, M., Hussain, O.K., 2020. Intellibot: A dialogue-based chatbot for the insurance industry. Knowledge-Based Systems 196, 105810.

[46] Pamungkas, E.W., 2019. Emotionally-aware chatbots: A survey. arXiv preprint arXiv:1906.09774 .

[47] Pilato, G., D'Avanzo, E., 2018. Data-driven social mood analysis through the conceptualization of emotional fingerprints. Procedia computer science 123, 360–365.

[48] Poria, S., Majumder, N., Mihalcea, R., Hovy, E., 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances. IEEE Access 7, 100943–100953.

[49] Rashkin, H., Smith, E.M., Li, M., Boureau, Y.L., 2018. Towards empathetic open-domain conversation models: A new benchmark and dataset. arXiv preprint arXiv:1811.00207 .

[50] Rollo, F., Bonisoli, G., Po, L., 2022. Supervised and unsupervised categorization of an imbalanced italian crime news dataset, in: Information Technology for Management: Business and Social Issues: 16th Conference, ISM 2021, and FedCSIS-AIST 2021 Track, Held as Part of FedCSIS 2021, Virtual Event, September 2–5, 2021, Extended and Revised Selected Papers, Springer. pp. 117–139.

[51] Santhanam, S., Shaikh, S., 2019. A survey of natural language generation techniques with a focus on dialogue systems-past, present and future directions. arXiv preprint arXiv:1906.00500 .

[52] Scherer, K.R., Wallbott, H.G., 1994. " evidence for universality and cultural variation of differential emotion response patterning": Correction. .

[53] Schuff, H., Barnes, J., Mohme, J., Padó, S., Klinger, R., 2017. Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus, in: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 13–23.

[54] Shmueli, B., Ku, L.W., 2019. Socialnlp emotionx 2019 challenge overview: Predicting emotions in spoken dialogues and chats. arXiv preprint arXiv:1909.07734 .

[55] Varshney, D., Ekbal, A., Bhattacharyya, P., 2021. Modelling context emotions using multi-task learning for emotion controlled dialog generation, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 2919–2931.

[56] Vogt, J., 2021. Where is the human got to go? artificial intelligence, machine learning, big data, digitalisation, and human-robot interaction in industry 4.0 and 5.0. AI Soc. 36, 1083–1087.

[57] Wang, Y., 2020. Iteration-based naive bayes sentiment classification of microblog multimedia posts considering emoticon attributes. Multim. Tools Appl. 79, 19151–19166.

[58] Wani, A.H., Hashmy, R., 2023. A supervised multinomial classification framework for emotion recognition in textual social data. Int. J. Adv. Intell. Paradigms 24, 173–189.

[59] Young, T., Cambria, E., Chaturvedi, I., Zhou, H., Biswas, S., Huang, M., 2018. Augmenting end-to-end dialogue systems with commonsense knowledge, in: Proceedings of the AAAI conference on artificial intelligence.

[60] Zad, S., Heidari, M., James Jr, H., Uzuner, O., 2021. Emotion detection of textual data: An interdisciplinary survey, in: 2021 IEEE World AI IoT Congress (AIIoT), IEEE. pp. 0255–0261.

[61] Zhang, Y., Sun, S., Galley, M., Chen, Y.C., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, B., 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. arXiv preprint arXiv:1911.00536 .

[62] Zhong, P., Wang, D., Miao, C., 2019. Knowledge-enriched transformer for emotion detection in textual conversations. arXiv preprint arXiv:1909.10681 .

Journal of Visual Language and Computing

Volume 2023, Number 1