An Agile Conflict-Solving Framework for Intent-Based Management of Service Level Agreement

Nicollas R. de Oliveira[‡], Igor M. Moraes[‡], Dianne Scherly V. de Medeiros[‡],
Martin Andreoni Lopez[*], Diogo M. F. Mattos[‡]
[‡]LabGen/MídiaCom – PPGEET/TCE/IC – Universidade Federal Fluminense – UFF – Niterói, Brazil
[*]Technology Innovation Institute, 9639 Masdar City, Abu Dhabi, UAE

***Abstract*—Managing and orchestrating Radio Access Networks (RAN) are error-prone tasks due to the high number of parameters and the complex interaction among all the network components. This paper proposes the AGIR system (AGility in Intent-based management of service-level agreement Refinements) for Open Radio Access Networks (Open RAN). AGIR implements intention-based network management, allowing operators to specify Service-Level Agreements (SLAs) for the RAN to fulfill. The system translates imprecise intentions into configurable network instructions, enhancing flexibility, scalability, and reducing human errors. The proposal enhances productivity, lowers operating costs, improves user experience, and optimizes network performance through real-time data analysis and automation. AGIR addresses the need for open and comprehensive interfaces in RANs, enabling cooperation among components from different suppliers. Our results reveal that the proposed system reaches more than 80% precision in detecting conflicting intentions when deploying a deep neural network based on Long Short-Term Memory with 256 neurons.**

***Index Terms*—Mobile Networks, Cellular Communications, Next-Generation Networks, 6G**

I. Introduction

Current Radio Access Networks (RANs) deployments rely on closed hardware and software solutions, making network operators and service providers dependent on proprietary solutions [1]. Aiming at improving flexibility, increasing interoperability, and adding new functionalities to the RAN, the telecommunications industry created an initiative called Open RAN [2]. The goal is to modify the network architecture to enable operators to deploy multi-vendor solutions for RANs using non-proprietary components. It is possible due to hardware and software disaggregation, open and interoperable interfaces, and virtualization [3]. In addition to increasing the autonomy of operators to create solutions that do not rely on a single vendor, Open RAN also focuses on adding intelligence and enabling RAN programmability. Defining a new architecture based on open standards is fundamental to achieving the Open RAN goals. To this end, the O-RAN Alliance was created in a standardization effort. The O-RAN Alliance is an international consortium comprising over 30 operators and over 200 telecommunications vendors. The O-RAN Alliance specifications provide detailed schemes for building RAN solutions that meet the Open RAN requirements, making network orchestration and management more efficient. This scenario favors the creation of more reliable, secure, and high-performing networks capable of operating autonomously, governed by verifiable closed control loops [4].

RAN orchestration relies on implementing complex policies, typically described as high-level goals or business intentions in current mobile communication networks. The high-level goals are represented by Key Performance Indicators (KPIs), which allow managers to track the progress of operations while abstracting the specifics of network management and operation. Operators then perform the complex and error-prone task of decomposing each high-level goal into low-level actions to be deployed on relevant physical or virtual devices [5]. Intentions express the operator's expectations regarding network and service management. An intention consists of a set of operational Service Level Objectives (SLO) the network should achieve and results that the network should deliver, defined declaratively without specifying how to achieve or implement them [6]. The objectives and expected behavior of the network can be defined by one or multiple intentions [7]. In this context, current-technology RAN challenges the operators to orchestrate and translate high-level business goals into low-level network policies and actions.

Open RAN leverages intent-based management to automate network configuration and monitoring. Intent-based management involves declaring high-level intentions that define the network behavior according to the operators' specifications. These specifications are provided through goals or KPIs without the need for the network to be explicitly programmed to achieve the service level objectives. By applying intent-based management to the RAN, the network configuration can be transformed from fine-tuning technical parameters to high-level definitions, allowing service providers to specify the connectivity services based on business intentions. Although intentions can be expressed through declarative models, this approach requires high precision in expressing policies, as it can lead to ambiguities. Network administrators are essential in eliminating ambiguity in policy expression when using declarative models [8]. Using natural language to express

This work is partially funded by CNPq, CAPES, FAPERJ, FAPESP (2018/23062-5), City Hall of Niterói/FEC/UFF (PDPA 2020), and RNP.

intentions facilitates goal description for non-technical individuals or those unfamiliar with technical jargon in a specific domain. However, natural language is also prone to ambiguity, which makes it challenging for the system to capture the operator's intention unambiguously and accurately. Despite this, ambiguity can be reduced when using natural language, as it allows adding context and details to the intention description [8]. However, for this description to be used as input in intent-based management systems, it must be processed to extract facts and indicators. After this procedure, the necessary actions to achieve management objectives are inferred [9].

Natural Language Processing (NLP), or computation linguistics, can extract facts and indicators. NLP is a research field involving computational models and processes for solving practical problems related to understanding and manipulating human languages [10]. Expressing intentions directly in natural language allows for the abstraction of management interfaces across different devices. Thus, it becomes possible to reduce the likelihood of human-prone errors when manually translating policies into equipment configuration commands. However, intent-based management systems do not guarantee the network's sustainability, as they may not encompass all possible situations that may arise.

We propose the AGility in Intent-based management for service level Refinement (AGIR) system, which implements intent-based network management capable of detecting conflicts between intentions and policies, mitigating them, and providing the agreed service level for the RAN. The AGIR system refines the network policy according to the intentions extracted from the Service Level Agreements (SLAs). The system receives intentions and network monitoring data and sends configuration and optimization instructions to the network entities. The system design follows a modular architecture, consisting of four modules to execute the process: Intelligent Application (iApp), Translator, Conflict Resolver, and Network Agent.

In this paper, we focus on the Conflict Resolver module. We deploy and evaluate a recurrent deep-learning model for Natural Language Processing. Our results reveal that intention declarations show recurrent disassortative patterns, and some network entities are likelier to be the subject of an intention than others. These results evidence that conflict identification should consider the recurrent behavior of the intention declarations. Thus, the proposed conflict identification mechanism achieved a precision higher than 80% when deploying a Long Short-Term Memory (LSTM) neural network.

The remainder of this paper is organized as follows. Section II discusses the related work. Section III presents an overview of the O-RAN Architecture and intent-based management. Section IV presents the architecture proposed for the AGIR system, detailing the Conflict Resolver module. In Section V, we discuss the results. Finally, Section VI concludes this paper and presents future work.

II. Related Work

Deploying network policies challenges operators because it requires translating high-level policy goals into low-level tasks into network-specific-device configuration syntax. Intent-Based Networking (IBN) aims to address this issue by allowing operators to specify high-level policies without the burden of dealing with network configuration or programming interfaces [11]. Expressing intentions in natural language offers several benefits, which include avoiding learning new programming languages and reducing human-prone errors in policy translation. However, the flexibility also introduces challenges in generating unambiguous and accurate configurations that precisely capture the operator's intent. Despite the potential of IBN for fast, automated, and reliable policy deployment, the current limitations regarding the direct use of natural language policy documentation in intent-based management systems impose limitations on the current technology. In turn, previous work surveys the usage of Large Language Models (LLM) to enhance the network management [12], [13]; introduces a multi-agent generative artificial intelligence network that exploits the collective intelligence to generate an LLM [14]; and also focuses on providing a chatbot capable of translating natural language intentions into an abstract artificial language that unequivocally describes the intention [5].

In the context of IBN, Clemm *et al.* define that a service model describes instances of services offered to customers and relies on lower-level models (device and network models), which facilitates the service mapping onto the underlying network or information technology infrastructure [11]. While system orchestration is necessary to instantiate a service model, the logic for managing and mapping it is not embedded. On the other hand, a policy comprises rules that express simple control loops and can be directly implemented by devices without requiring intervention from external systems. These policies dictate the actions to be taken under specific events or conditions but do not explicitly specify the desired outcome. In contrast, an intent operates at a broader scope, encompassing the network and service level, to define high-level operational goals and outcomes without providing an exhaustive enumeration of specific events or actions. The intent system can autonomously derive the algorithms or rules necessary to achieve the intended outcomes. In an autonomic networking context, the network is ideally responsible for rendering the intent, disseminating it, and coordinating actions between nodes without relying on external systems.

Jacobs *et al.* propose LUMI, a language model designed explicitly for university campus networks, offering operators the capability to communicate their intentions to the network using natural language. LUMI's key functionality involves translating natural language expressions into a precise network configuration abstraction layer, enabling the seamless implementation of the operator's intention within the network infrastructure. LUMI deploys a chatbot and relies on Named Entity Recognition (NER) algorithms to accurately extract and label entities from the operator's input in natural language.

Moreover, LUMI also introduces the Network Intent Language (Nile), an intermediary abstraction layer, acting as an intermediary syntax between natural language expressions and the corresponding network configuration instructions. LUMI leverages the *word2vec* algorithm for word embedding and employs the Bi-LSTM (Bidirectional Long Short-Term Memory) model to enhance contextual understanding, evaluating phrases in both left-to-right and right-to-left directions. The authors evaluate LUMI performance by assessing its information extraction accuracy and learning capabilities through operator feedback and measuring compilation and deployment times across various campus network topologies. The proposal evaluation deploys diverse datasets comprising synthesized intentions and real-world intention data derived from 50 distinct campus networks in the United States of America.

Mattos, Duarte, and Pujolle propose the Reverse Update mechanism, which introduces a practical approach for updating rules in software-defined networks [15]. The mechanism involves tagging policy versions in the network and updating forwarding rules in reverse along the established flow during ongoing policy updates. The authors emphasize the significance of identifying conflicts between policies, which can be either partial or total, based on the policy application domain. Accurate identification of conflicts, especially partial conflicts, is crucial as they may result in contradictory forwarding rules in the network, demanding the rejection of one of the policies.

Unlike previous work, the proposed AGIR system utilizes conflict identification to assess whether proposed intentions can be accepted or must be rejected due to conflicts with already implemented policies in the network. A critical factor in the AGIR system is the conflict identification process that must correctly identify the domain affected by an intention proposal. In this paper, we characterize the natural language intentions and propose a deep learning model that precisely identifies conflicts among intention proposals.

III. O-RAN Architecture and Intent-based Management

The O-RAN architecture specified by the O-RAN Alliance is shown in Figure 1. The specification follows the disaggregation of gNodeB's three functional units proposed by 3GPP and introduces the O-RAN Central Unit (O-CU), O-RAN Distributed Unit (O-DU), and O-RAN Radio Unit (O-RU). The O-RU is a logical node hosting the lower physical layer (*Low-PHY*) functions and radio frequency signal processing. The O-DU is a logical node hosting higher physical layer (*High-PHY*) functions, Medium Access Control (MAC), and Radio Link Control (RLC) sublayers. The O-CU is responsible for mobility control, RAN sharing, session management, and user data transfer [16].

The O-RAN architecture introduces the concept of RAN Intelligent Controllers (RICs), comprising the Non-Real-Time RIC (Non-RT RIC) and Near-Real-Time RIC (Near-RT RIC) [17]. These controllers offer a centralized view of the network, assess performance metrics, and apply machine learning algorithms for automated optimization tasks, such as network slicing, load balancing, and handovers. The Near-RT RIC, located at the network edge, operates closed-loop controls with periodicity between 10 milliseconds and 1 second. The Non-RT RIC complements the Near-RT RIC for intelligent and optimized RAN operation on a timescale greater than 1 second. The O-RAN architecture does not define the real-time control, i.e., for a periodicity smaller than 10 milliseconds. The Near-RT RIC comprises applications called xApps, and the necessary services for their execution. The xApp is a microservice that manages radio resources through standardized interfaces and service models. The Non-RT RIC supports third-party applications, called rApps, which provide value-added services, facilitating RAN optimization and operation [16], [17].

The O-Cloud is the RAN cloud infrastructure, consisting of infrastructure components that execute the necessary functionalities, such as O-CU Virtual Network Functions (VNFs) and the Non-RT RIC rApps [16]. To add flexibility, the O-RAN architecture specifies open interfaces, such as A1, E2, Open Fronthaul, O1, and O2, and also uses 3GPP interfaces, such as E1, F1, X2, Xn, NG, and Uu. Each interface interconnects specific components. The O1 interface serves for communication between the Service Management and Orchestration (SMO) framework and other components of the O-RAN architecture. For example, SMO uses the O1 interface to communicate with the Near-RT RIC. The communication between SMO and O-Cloud is performed via the O2 interface, enabling support for functionalities running in the cloud. The Non-RT RIC uses the A1 interface to send information to the Near-RT RIC, such as data about use cases and information enrichment. The E2 interface enables communication between the Near-RT RIC and managed elements, such as O-CU, O-DU, and O-eNB. The O-eNB is an Open-RAN-enabled eNodeB to provide LTE service. The Open Fronthaul interface allows interaction between O-RU and O-DU. Some components inherited from previous RAN generations use the same interfaces used in the architectures of those generations. An example is the E1 interface, which connects the O-CU's control and user planes. The F1 interface connects O-CU and O-DU elements for exchanging information about radio resource sharing and other network status. The X2 and Xn interfaces aid the interoperability between nodes of different generations, and the NG interface connects 5G nodes to the core network when the network operates in standalone mode, i.e., pure 5G [16]. The Uu interface allows interaction between the User Equipment (UE) and the O-RU or O-eNB.

Besides the facilitated interaction among components, RAN orchestration is still a complex task that relies on policies derived from high-level business goals or intentions. Such policies often describe the operator's expectations regarding network and service management and the KPIs, enabling managers to monitor operations without dealing with the intricacies of network management. To succeed in the orchestration, the operator needs to break down each policy into low-level actions to be deployed on relevant physical or virtual devices [18]. It is a complex and error-prone task in mobile com-

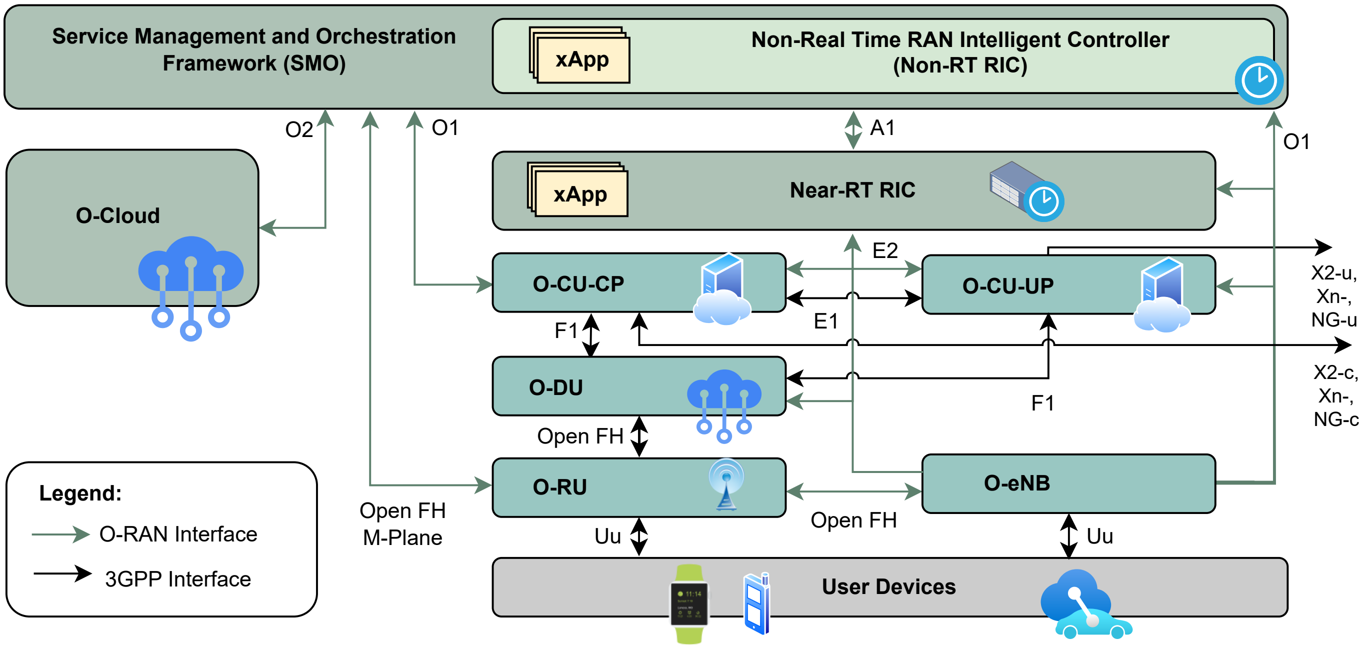


Fig. 1. The Open RAN architecture as specified by the O-RAN Alliance. The O-RAN Alliance standardized the open interfaces to allow communication between the O-RAN components. The 3GPP interfaces are used for communication between components inherited from other RAN generations.

munication networks, making RAN orchestration challenging for operators. The Open RAN relies on intent-based management to facilitate orchestration. Intent-based management involves high-level policies defining network behavior according to operator specifications, expressed through business goals or KPIs, without requiring explicit programming to achieve the service-level objectives. However, network policies are usually written in plain language. Therefore, intelligent processing is needed to extract facts and indicators to infer the necessary actions for management objectives to use them as input for intention-based management systems. In this scenario, NLP is paramount, as it involves computational models and processes for understanding and manipulating natural languages [19]. Regardless of its manifestation in text or speech, natural language refers to daily human communication. This definition excludes programming languages and mathematical notations, considered artificial languages. Expressing intentions directly in natural language helps abstract management interfaces across different equipment, but natural languages constantly evolve, hardening to establish explicit rules for computers [10]. Moreover, expressing intentions directly in natural language may introduce ambiguity, increasing the challenge for systems to capture operators' intentions unambiguously and accurately. Intention-based management systems may not guarantee network sustainability since they may only account for some possible scenarios. Knowledge management systems have been proposed as a countermeasure to facilitate decision-making processes [20]. Thus, a step toward a knowledge management system is to deploy a technique that identifies the affected domain by an intention proposal and compares the domain among all accepted intentions to warn whether an intention proposal results in a conflict with already established intentions.

IV. AGIR Proposed Architecture

The AGIR system implements intent-based network management for the Open RAN, considering the architecture specified by the O-RAN Alliance. The system follows a modular architecture, as shown in Figure 2, and is based on NLP to extract information from intentions. The iApp module serves as the entry point for the intentions expressed by the operator. It envisions a simple graphical interface for the application users to interact with and input their intentions. In addition to being the users' point of contact, the iApp is also responsible for forwarding the received intentions to the Translator module, receiving network monitoring data, constantly validating if the created policies are suitable to achieve the goals established in the intentions, and indicating to the Translator module the need for adaptation in the mapped policies.

The Translator module transforms the intention expressed in natural language into configuration statements. The module implements NLP algorithms to translate the intention, converting it into a policy to be implemented through a Software-Defined Networking (SDN) controller [21]. Intention translation is based on the lexical and morphological analysis of textual entities. It allows for identifying keywords that compose the intention, which can then be used to generate a rule model that can be transformed into a policy. The intention's keywords are compared with a pre-existing knowledge base in the network to classify them according to the labels of key elements in the model. The model is an expression of the

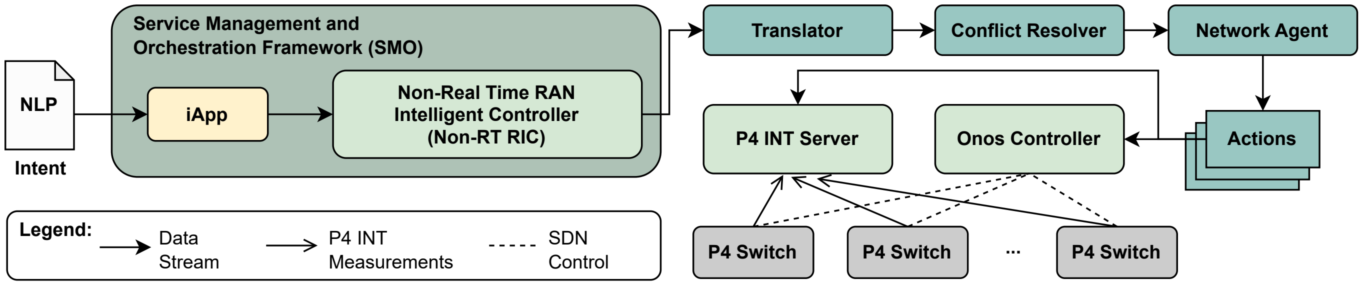


Fig. 2. Block diagram of the AGIR system architecture integrated with elements of the architecture specified by the O-RAN Alliance.

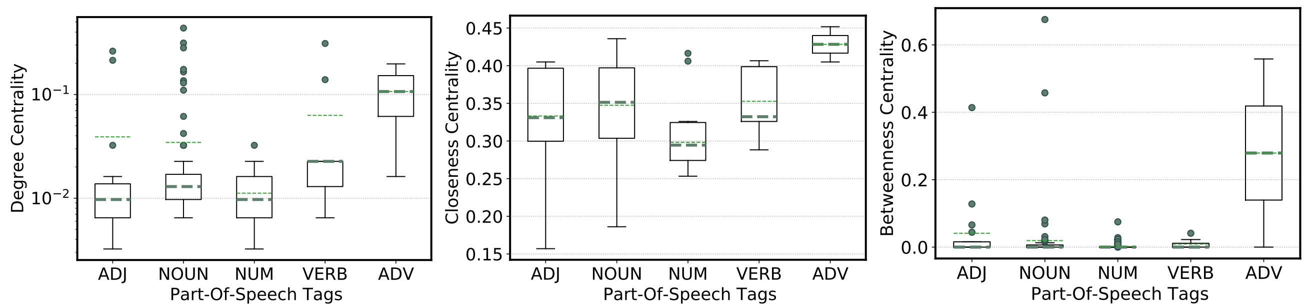
intention presented in an object-action-result format. The idea is to formulate strategies, which are the actions for the current network resources, which are the objects of the model, to achieve the intention's goal, which is the result. By converting the keywords into structured statements, the SDN controller's translation into policies to be executed becomes easier.

Internally, translating intentions can follow a sequence of text-cleaning and shaping techniques. Among these techniques are (i) tokenization, (ii) removal of punctuation, special characters, and stopwords, (iii) spelling correction, (iv) recognition of specific named entities, and (v) stemming or lemmatization. Guided by the mentioned order, the text that makes up each intention is first subjected to tokenization, a discretization procedure. Using the space character as a delimiter criterion, for example, tokenization transforms each contiguous sentence of an intention into a list of tokens, allowing for individual manipulation. Each token is seen as an instance of a character sequence. Next, orthographic features such as punctuation and special characters are removed from each token. Since they do not contribute to the semantic understanding of the intention, stopwords, such as conjunctions, articles, and pronouns, are also removed from the text. To deal with possible typographical errors, spelling correction is advisable, a procedure performed by comparing each token with its closest counterpart in a customized dictionary. Named Entity Recognition (NER) aims to identify names of software, hardware, or any proper names related to the intention. Finally, adopting lemmatization or stemming is common to reduce unnecessary processing caused by potential redundancies between words, either through inflections or derivations. In lemmatization, the goal is to eliminate possible variants or plurals of the same word, reducing them to the same lemmas, known as dictionary forms. In contrast, this reduction is done in stemming by transforming each word into its root. However, lemmatization or stemming are optional steps in intention processing due to intention declarations being a closed domain of knowledge and agnostic to variations of a word root.

Expanding textual processing to other linguistic stages, some NLP techniques can perform morphosyntax analysis in different degrees of complexity. At a fundamental tier, POS (*part-of-speech*) tagging is characterized as a morphological analysis technique that returns only the lowest layer of the parse tree, i.e., the grammatical tagging. Thus, each word in a sentence is assigned metadata, identifying its grammatical classification and conjugational attributes [10]. Adopting the techniques mentioned above enables them to convert an intention described in natural language, which may be imprecise and ambiguous, into an interpretable action by the network.

The Network Agent module communicates the policies obtained from the translated intentions to the SDN controller. The ONOS controller is considered for network control, communicating to the Network Agent through the northbound interface. P4 switches are not explicit components of the AGIR system but are used to provide information to the proposed system. These switches are responsible for In-band Network Telemetry (INT), providing network information to the controller and a P4 telemetry server (INT P4 Server). This server sends measurement reports to the iApp, allowing for continuous network performance monitoring. The optimization of network performance will be carried out using machine learning mechanisms such as Deep Reinforcement Learning (DRL) or Fuzzy Reinforcement Learning (FRL) [22]. It enables the validation of whether the goals of the intention are being achieved and the adaptation of policies if necessary. It is worth noting that the proposed system is related to the Non-RT RIC, meaning that the control loop governing the process lasts longer than 1 second.

The Conflict Resolver module is responsible for detecting and mitigating conflicts and is the focus of this paper. The operator's intentions can conflict, creating contradictions when the specified goals require modifications to the same parameter but in different quantities [7]. The existence of these conflicts makes intent-based RAN management hard. The conflict resolution in the AGIR system is based on the methodology of reverse updates, a policy update scheme for SDN that ensures consistency in policy commitments. Reverse update relies on updating the flow processing and forwarding rules in the reverse direction of the already installed flow path, ensuring that a flow always reaches the most up-to-date network configuration [15]. It is important to highlight that the step before the conflict resolution is identifying the conflict. The AGIR system utilizes conflict identification to assess whether proposed intentions can be accepted or must be rejected due to conflicts with already implemented policies in the network. A critical factor in the AGIR system's conflict identification process is correctly identifying the domain



(a) Degree Centrality.

(b) Closeness Centrality.

(c) Betweenness Centrality

Fig. 3. Results of applying three centrality metrics on the word graph generated from the preprocessed network intentions. Among the different groups of grammatical classes, adverbs demonstrate a more central behavior regardless of the evaluated metric.

affected by the intention proposal. In essence, this module approaches the conflict identification task as a classification problem based on deep learning algorithms. Thus, we model a neural network architecture comprising Embedding, LSTM (Long Short-Term Memory), and Dense layers using the Keras[1] library. The embedding layer performs a mapping of *tokenized* sentences to dense, fixed-size arrays using a pre-trained *Word2Vec* model. This word embedding technique effectively captures the semantic meaning and relationships between the meaningful word representations injected into the next LSTM layer. The second layer comprises an LSTM model, an algorithm capable of learning long-range dependencies and sequential patterns in data. This ability to maintain and update cell states ensures that LSTM retains relevant information over different intervals, making it especially suited for text-related tasks. Composed of only one neuron, the Dense layer adopts a Sigmoid activation function. The choice is justified because the Sigmoid function is especially suitable for problems involving binary classification, making it possible to map the output to probabilities between 0 and 1. The model is also trained using the Adam optimizer over the binary cross-entropy loss function. In practice, the neural network model aims to analyze and classify the existence of semantic conflicts between sentences, making it a valuable tool for automating the process based on intentions written in a high-level language.

V. Results and Discussion

We evaluate the proposed conflict-identification mechanism of the Conflict Resolver module. The evaluation was conducted by adopting one of the datasets synthetically built by Jacobs *et al.* [5]. Such a dataset comprises 10k pairs of network intentions duly balanced and labeled between conflicting or not labels. Each intent, written in natural language, contains the actions and parameters instructed by the network operator, covering tasks such as router configuration, traffic monitoring, fault diagnosis, and performance optimization. The algorithms and techniques described in this paper are developed in Python using a personal computer equipped with an Intel Core i7 4770 processor, with 8 GB of RAM and 1 TB of storage. It is noteworthy that the presented results are mean values with a confidence interval of 95%.

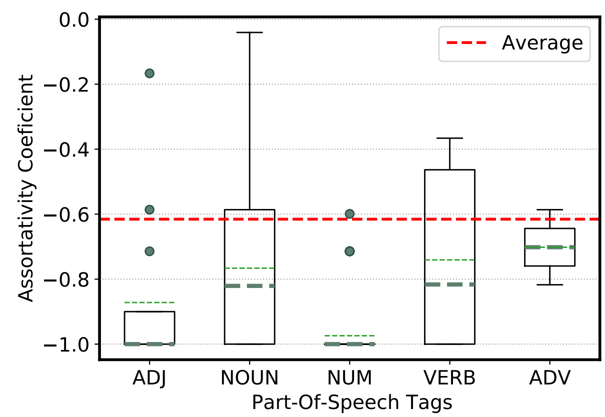


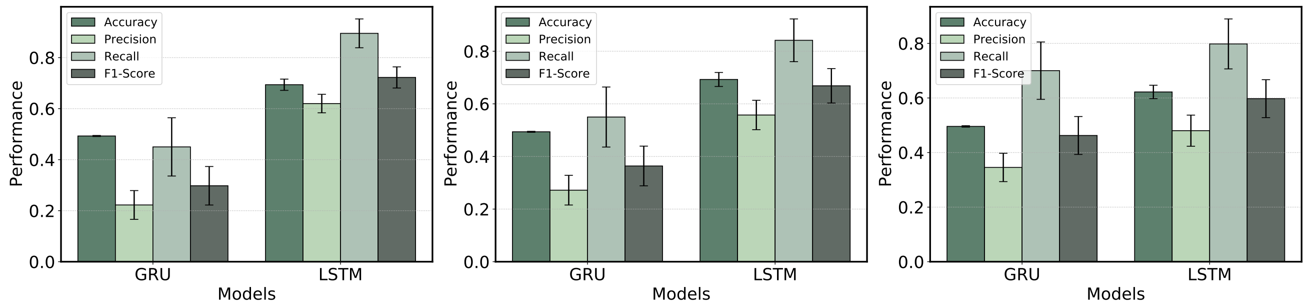
Fig. 4. Results reveal a recurrent disassortative pattern ($A_c < 0$) across all grammatical classes, indicating central words mostly link to low-degree words.

The proposal implements a comprehensive text preprocessing pipeline containing different functions from the NLTK[2] library. The objective is to reduce the plurality of words, ensuring that only the essential and most informative words are ingested in subsequent steps. Headed by tokenization, preprocessing starts by converting the contiguous sentences into lists of tokens and removing punctuation and special characters. However, the existence of IP addresses and measurement units imposes the need to detect and preserve such specific structures since they are parameters commonly passed in intentions. This need is addressed by integrating regular expression functions. Once such structures are identified, POS Tagging and stopword filtering can be applied. This order of implementation seeks to mitigate potential labeling errors that occur when parsing extremely truncated sentences, i.e., without connectives, articles, or pronouns.

In practice, the evaluation process focuses on two main approaches, one from a graph perspective and the other from an algorithmic perspective. The first approach envisages the structuring of an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

[1]https://keras.io/.

[2]https://www.nltk.org/.



(a) 64 neurons.
(b) 128 neurons.
(c) 256 neurons.

Fig. 5. The second evaluation approach involves the performance comparison of different neural network models during semantic conflict detection between network intentions. Results are presented through information retrieval metrics across different hidden layer neuron configurations.

based on preprocessed intentions and converted into vectors of meaningful tokens. In this graph, each node represents a distinct token, and the edge between a pair of nodes contains a weight proportional to the frequency of co-occurrence of the respective tokens in the corpus of intentions. Analyzing the graph generated through centrality or structural metrics, it is possible to reveal several imperceptible characteristics in the textual format. Although all centrality metrics aim to quantify the importance of each node in a graph, each metric associates importance with a given feature. The analyzed classic centrality metrics are degree, closeness, and betweenness. Degree centrality calculates the importance of each node considering its degree, that is, the number of edges connected to each node. The degree centrality of a node υ_i is represented by

$$C_{deg}(\upsilon_i) = deg(\upsilon_i). \tag{1}$$

Similarly, closeness centrality relates to how quickly a node reaches all other nodes in the network. The calculation of closeness centrality ($C_C(\upsilon_i)$) of each node υ_i takes into account the shortest paths between υ_i and all other nodes in the network. Mathematically, the closeness is given by

$$C_C(\upsilon_i) = \frac{|\mathcal{V}| - 1}{\sum_{j \neq i} \delta^*(\upsilon_i, \upsilon_j)}, \tag{2}$$

where $|\mathcal{V}|$ is the total number of nodes and $\delta^*(\upsilon_i, \upsilon_j)$ is the shortest distance, in number of hops, between the pair of nodes υ_i and υ_j. Naturally, the lack of a connection between υ_i and υ_j implies a distance $\delta^*(\upsilon_i, \upsilon_j) = \infty$. In contrast, betweenness centrality reflects the total fraction of shortest paths that passes through a node, using it as a bridge. Formally, the betweenness centrality (C_B) of a node υ_i is defined as

$$C_B(\upsilon_i) = \sum_{\upsilon_s \neq \upsilon_t \neq \upsilon_i \in \mathcal{V}} \frac{\sigma_{st}(\upsilon_i)}{\sigma_{st}}, \tag{3}$$

where $\sigma_{st}(\upsilon_i)$ represents the number of shortest paths from node υ_s to node υ_t that pass through node υ_i and σ_{st} is the total shortest paths from node υ_s to node υ_t. Thus, the ratio represents the proportion of shortest paths between υ_s and υ_t that passes through υ_i. In an intent graph, the betweenness centrality exposes insights into critical words or terms that lead to intentions containing unusual parameters or actions.

Additionally, we evaluate the behavior of the graph from the perspective of assortativity (A_c), a structural metric defined between $[-1.1]$. This metric expresses the tendency of nodes to connect to other nodes with similar values of a given characteristic. For example, when considering the degree as a characteristic to be evaluated, positive assortativity values indicate a correlation between nodes of similar degrees. In contrast, negative values indicate relationships between nodes of different degrees. Values close to zero translate the complete connection between all nodes in a graph. Extreme positive or negative cases show that the graph exhibits mixing patterns between perfect orders or unordered patterns, respectively.

Figures 3(a), 3(b), and 3(c) show the degree, closeness, and betweenness centrality metrics. Figures show boxplots for each group of grammatical class words present in the graph of intentions. Regardless of the type of centrality, the words or terms classified as numerals have low centrality values. This phenomenon happens because numerical values used in the intentions are particular and precise since they typically comprise parameters such as IP addresses and bandwidth values. The fact places numerical-value nodes at the graph edge, making them less central. As seen in Figure 4, a recurrent disassortative pattern ($A_c < 0$) is observed in all groups of grammatical classes (POS tags), showing that most of the nodes from central words come from words with lower degrees. Another interesting finding is that several nodes are classified as nouns, demonstrating similarity to a complete mesh, that is, connected to all the graph nodes. This fact is corroborated by Figure 3(a), as nouns have higher degree centrality values ($C_{deg} \approx 0$).

The second evaluation approach explores the algorithmic perspective, comparing the performance of different neural network models and parameters. The evaluation considers two deep learning algorithms, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Both algorithms were trained using 80% of the intentions dataset and then tested with the rest of the samples to detect semantic conflict between pairs of intentions. Figures 5(a), 5(b), and 5(c) depict the values of the information retrieval metrics, i.e., accuracy, precision, recall, and f1-score, when classifying pairs of intentions. The figures show neural network configurations with 64, 128, and 256 neurons in the hidden layer. The analysis of the results for each algorithm in each scenario points out a consensus that the results of the hierarchical grouping were the best. Especially in tests adopting 64 neurons, a superior performance of the LSTM model is observed compared to the GRU model in all evaluated metrics. Although this performance gap decreases in tests adopting 128 neurons, it returns considerably in the test with the maximum number of neurons, i.e., 256. However, it is worth noting that the LSTM algorithm consistently provides high-precision models for detecting conflicts among intentions, over 80% precision, regardless of neuron quantity in the hidden layer.

VI. Conclusion

This paper introduces the AGility in Intent-based management for service level Refinement (AGIR) system, designed to address the complexities of Open Radio Access Networks (Open RAN). AGIR presents a cutting-edge approach to intention-based network management, empowering operators to define and enforce precise Service-Level Agreements (SLAs) within the RAN environment. The system acts as a bridge between intentions and actions, seamlessly translating high-level operator intentions from the SLAs into actionable network instructions. This transformative capability enhances the network's flexibility and scalability and reduces potential human-prone errors. Comprising four integral modules, Intelligent Application (iApp), Translator, Conflict Resolver, and Network Agent, the AGIR system operates cohesively to execute its core process. The evaluation process focused mainly on the Translator and Conflict Resolver modules in a dual approach. By building a graph structure based on analyzed intentions, the first approach compares the centrality and assortativity metrics of each group of parts-of-speech tags. The second approach compared the performance of two deep neural network models, LSTM and GRU, submitted to classify intentions as conflicting. Therefore, the superiority of the LSTM model was verified in all scenarios and tested metrics. LSTM models reached more than 80% precision in classifying conflicting intentions. In future work, we intend to test other neural network models and improve the recognition of named entities related to the scope of networks based on training with corpora of realistic and more populated intentions.

References

[1] M. Andreoni, G. N. N. Barbosa, and D. M. F. Mattos, "New barriers on 6g networking: An exploratory study on the security, privacy and opportunities for aerial networks," in *2022 1st International Conference on 6G Networking (6GNet)*, 2022, pp. 1–6.

[2] R. S. Couto, D. M. Mattos, I. M. Moraes, P. H. C. Caminha, D. S. V. de Medeiros, L. A. C. de Souza, M. E. M. Campista, and L. H. M. K. Costa, "Gerenciamento e orquestração de serviços em o-ran: Inteligência, tendências e desafios," *Minicursos do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2023.

[3] D. M. F. Mattos and O. C. M. B. Duarte, "Xenflow: Seamless migration primitive and quality of service for virtual networks," in *2014 IEEE Global Communications Conference*, 2014, pp. 2326–2331.

[4] N. Foster, N. McKeown, J. Rexford, G. Parulkar, L. Peterson, and O. Sunay, "Using deep programmability to put network owners in control," pp. 82–88, 2020.

[5] A. S. Jacobs, R. J. Pfitscher, R. H. Ribeiro, R. A. Ferreira, L. Z. Granville, W. Willinger, and S. G. Rao, "Hey, lumi! using natural language for {intent-based} network management," in *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, 2021, pp. 625–639.

[6] A. Clemm, M. F. Zhani, and R. Boutaba, "Network management 2030: Operations and control of network 2030 services," *Journal of Network and Systems Management*, vol. 28, no. 4, pp. 721–750, 2020.

[7] A. Banerjee, S. S. Mwanje, and G. Carle, "Contradiction management in intent-driven cognitive autonomous ran," in *2022 IFIP Networking Conference (IFIP Networking)*. IEEE, 2022, pp. 1–6.

[8] N. Race, I. Eckley, A. Parlikad, C. Rotsos, N. Wang, R. Piechocki, P. Stiles, A. Parekh, T. Burbridge, P. Willis *et al.*, "Industry-academia research toward future network intelligence: The NG-CDI prosperity partnership," *IEEE Network*, vol. 36, no. 1, pp. 18–24, 2022.

[9] N. R. de Oliveira, D. S. Medeiros, and D. M. Mattos, "A syntactic-relationship approach to construct well-informative knowledge graphs representation," in *2020 4th Conference on Cloud and Internet of Things (CIoT)*. IEEE, 2020, pp. 75–82.

[10] N. R. de Oliveira, P. S. Pisa, M. A. Lopez, D. S. V. de Medeiros, and D. M. Mattos, "Identifying fake news on social networks based on natural language processing: trends and challenges," *Information*, vol. 12, no. 1, p. 38, 2021.

[11] A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura, "Intent-Based Networking - Concepts and Definitions," Internet Engineering Task Force, Internet-Draft draft-irtf-nmrg-ibn-concepts-definitions-00, 2019, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-irtf-nmrg-ibn-concepts-definitions/00/

[12] L. Bariah, Q. Zhao, H. Zou, Y. Tian, F. Bader, and M. Debbah, "Large language models for telecom: The next big thing?" *arXiv preprint arXiv:2306.10249*, 2023.

[13] B. Brik, H. Chergui, L. Zanzi, F. Devoti, A. Ksentini, M. S. Siddiqui, X. Costa-Pérez, and C. Verikoukis, "A survey on explainable AI for 6G O-RAN: Architecture, use cases, challenges and research directions," *arXiv preprint arXiv:2307.00319*, 2023.

[14] H. Zou, Q. Zhao, L. Bariah, M. Bennis, and M. Debbah, "Wireless multi-agent generative AI: From connected intelligence to collective intelligence," 2023.

[15] D. M. F. Mattos, O. C. M. B. Duarte, and G. Pujolle, "Reverse update: A consistent policy update scheme for software-defined networking," *IEEE Communications Letters*, vol. 20, no. 5, pp. 886–889, 2016.

[16] A. Arnaz, J. Lipman, M. Abolhasan, and M. Hiltunen, "Toward Integrating Intelligence and Programmability in Open Radio Access Networks: A Comprehensive Survey," vol. 10, pp. 67 747–67 770, 2022.

[17] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," pp. 1–1, 2023.

[18] A. S. Jacobs, R. J. Pfitscher, R. H. Ribeiro, R. A. Ferreira, L. Z. Granville, W. Willinger, and S. G. Rao, "Hey, Lumi! Using Natural Language for Intent-Based Network Management," 2021, pp. 625–639.

[19] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," 2020.

[20] A. Leivadeas and M. Falkner, "A Survey on Intent Based Networking," pp. 1–32, 2022.

[21] D. M. F. Mattos, O. C. M. B. Duarte, and G. Pujolle, "A resilient distributed controller for software defined networking," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[22] R. H. Santos Filho, T. N. Ferreira, D. M. Mattos, and D. S. Medeiros, "An efficient and decentralized fuzzy reinforcement learning bandwidth controller for multitenant data centers," *Journal of Network and Systems Management*, vol. 30, no. 4, p. 53, 2022.