N-gram Based Croatian Language Network: Application in a Smart Environment

Renato Šoić, and Marin Vuković

Original scientific article

Abstract—**In the field of natural language processing, language networks represent a method for observing linguistic units and their interactions in different linguistic contexts. This paper uses the previously presented Croatian language network for building a solution capable of generating spoken notifications in Croatian language. The novelty of this paper is that it proposes an approach for generating spoken notifications in smart environments by combining specialized services that enable interaction with the environment and human users. The process employed for generating spoken notifications is described in detail. Also, a novel contribution of this paper is the case-study evaluation of the proposed approach in a smart home environment.**

Index terms—**n-grams, language network, natural language processing, smart environment.**

I. INTRODUCTION

In the last decade, natural language processing has made rapid progress. From the user's perspective, this is reflected in services and applications that provide reliable speech recognition and natural-sounding speech synthesis, as well as advanced natural language understanding and natural language generation. As a result, speech-enabled systems are being successfully deployed in a variety of domains and continue to improve significantly. There are several reasons for these improvements. Available computational power has increased, allowing the implementation of more efficient methods using large datasets. In addition, end-user devices have also evolved significantly: Powerful smartphones and various other smart devices are now widely available. Finally, the supporting infrastructure has also improved, enabling the aforementioned developments in natural language processing.

From the perspective of a minority language such as Croatian, there are still many challenges to making progress comparable to the services available for English. As far as speech technologies are concerned, Croatian language is still underdeveloped, although almost ten years have passed since the last thorough analysis [1].

The obvious reason for this situation is the fact that it is not relevant on a global level. Minority languages are rarely well developed in the field of speech technologies since there is no significant economic benefit. The other reason is related to linguistic complexity. Croatian language is morphologically very rich, which increases the complexity in many areas of natural language processing [2]. Methods used in the case of English language cannot be directly applied because Croatian language is different in terms of orthography, phonetics and morphology. Moreover, there are still no suitable open datasets for research purposes. A rare positive example is the n-gram system collected in scope of Hascheck, a Croatian online academic spell checking service [3][4].

N-gram based Croatian language network constructed using Hascheck's n-gram collection was previously presented by the authors in a conference paper [5]. Having a functional Croatian language network gives us the ability to build a system for spoken notifications in various environments. In order to do so, we should first define what is required for this purpose, mainly in terms of other building components, i.e. specialized services, resources and similar. Thus, the main novelty of this paper is the proposed approach of combining the previously presented Croatian language network with other required specialized services. In this sense, we build on top of the previously presented Croatian language network and provide additional explanation of its role in a broader context, where it represents only one, although crucial, component in a complex system.

The rest of the paper is organized as follows. In the following section, an overview of related research is given. Section III explains how our approach is applied in scope of a smart environment. Section IV describes the system architecture with corresponding linguistic services. In Section V the complete process of generating spoken notifications is explained. Section VI presents an evaluation on a use-case scenario in a smart home and discusses evaluation results. Finally, a conclusion is given, along with plans for future research and development.

II. RELATED WORK

With rapid advancements in speech technologies, spoken interaction has also become an important feature in smart environments. Enabling interaction with a computer system in the most natural way improved the quality of experience for consumers. As a result, speech technologies are used in many

different domains, such as smart homes [6], industrial facilities [7], smart vehicles [8], public institutions, etc. Very important is the benefit for people with complex communication needs. In some cases, interacting with a computer system in a conventional way was not possible at all, while interaction using speech opened new possibilities [9]. In the previously mentioned examples, spoken interaction is mostly limited to recognition of specific commands and reproduction of predefined, sometimes pre-recorded spoken notifications. However, the possibilities of spoken interaction between humans and computers have improved significantly with emersion of intelligent personal assistants. These systems have evolved to the point where they are capable of leading conversations with human users and understanding of linguistic and semantic context. Moreover, they can be integrated with various smart devices and systems, which increases their flexibility and expands their application areas [10]. Users can even develop new functionalities in scope of the given platform, which resulted with a constantly growing set of available services. The most advanced platforms rely on infrastructure in the computer cloud. This means that all sounds and conversations from the environment are being recorded, transmitted over the network and finally analyzed in the computer cloud [11].

To achieve comparable functionalities and quality of experience for the Croatian language, there are still many challenges. While there are commercial speech recognition and speech synthesis systems for the Croatian language, their performance is still not comparable in terms of accuracy and quality to the solutions available for the English language. This is not surprising, as these systems were originally developed for other languages and then adapted for Croatian language. Therefore, such speech synthesis solutions cannot achieve pronunciation quality that would please native speakers, while speech recognition systems are often unable to provide correct morphological forms in the recognition results. The academic community has conducted several research projects and incentives related to speech recognition [12][13] and speech synthesis [14]. However, these solutions are not ready for production and therefore are not available to the public.

Natural language generation in Croatian has already been applied in several scenarios from specific domains such as weather reports [15] and airline ticket sales [16]. However, these examples are of very limited scope and the employed methods would not be sufficient if applied in a smart environment. Other notable advances in the field of natural language generation and understanding for Croatian involve keyword extraction [17] and dialogue systems [15]. However, the techniques explored in these examples are not applicable to the underlying data structures in IoT platforms, but could be considered as improvements in the later stages of our research.

Previous work conducted by our research group was focused on smart environments and how their functionalities can be enhanced by integrating speech technologies. Regardless of the application area, smart IoT systems are distributed and offer different communication channels. Choosing the appropriate communication channel and interaction method is very important for human users. Therefore, we considered how to dynamically determine the interaction method used, depending on the conditions in the environment [18]. This highlighted the importance of understanding the environmental context and the opportunities this presents for optimizing devices and services in a smart environment [19]. This allowed us to define how natural language processing services can be integrated into the smart environment, performing in a distributed system setup [20]. The following research was focused on enabling spoken interaction capabilities in smart environments for the Croatian language. The first step in this complex and extensive research area were spoken notifications, applied NLP methods and their impact on user experience [21]. In previous research phases, we have also investigated how smart environments can be realized as distributed systems spread over an arbitrary number of system nodes [22]. In this case, we mainly focused on rule-based approach and monitoring, automation, and control in a smart environment, while spoken interaction had a limited scope.

III. CROATIAN LANGUAGE NETWORK IN A SMART ENVIRONMENT

Smart environments can be defined as physical spaces enriched with sensors and actuators, interaction interfaces and various services that can be embedded in specific devices or perform as standalone services on any generic device [23]. All elements of a smart environment are usually interconnected and communicate with each other directly or through an IoT platform. Smart environments are intended to be used by people. Therefore, the constant incentive is to support the most natural and intuitive interaction method for human users - speech [24].

Fundamental components for spoken human-computer interaction are speech recognition and speech synthesis. However, these components represent interaction interfaces but do not provide cognitive processing. Therefore, they are not sufficient to establish spoken interaction with a computer system or a smart environment. Additional components are required for this purpose. A natural language understanding component allows extracting meaning from captured user input that leads to an action being performed in the system [25]. A natural language generation component provides methods for creating meaningful content based on the data available in the system. This content is intended for a human user and therefore must be clear and concise, but also semantically and grammatically correct [26]. All the previously mentioned components are essential, and each of them requires a specific approach and methods from the field of natural language processing. In addition, many resources are needed for the implementation of the underlying models.

This paper focuses on spoken interaction and the methods required for natural language generation, linguistic resources and their relation to the environment. The novelty of this paper, in addition to the previously published work [5], is that here we describe the algorithmic approach used for inference of correct morphological forms in the domain of natural language generation. The solution proposed in this paper uses the n-gram based Croatian language network and related

prediction service from our previously published work [5] as one of the building blocks. Since our incentive was on enabling spoken interaction with a smart environment and designing a solution that can be integrated with systems from different domains, we decided to also investigate scenarios where the described solution is integrated with existing IoT platforms. Finally, the novel approach in this paper is evaluated in a smart home environment. Based on the evaluation results, we discuss how this approach can be used in real-time or near-real-time applications in a smart home environment.

Compared to modern natural language processing techniques based on machine learning and deep neural networks, the n-gram based approach cannot match the results achieved. N-gram based prediction models usually perform worse when compared to alternative methods applied to the same source corpora. However, the n-gram collection used in our research greatly exceeds the corpora available for training machine learning models. Hascheck is an online spell checking service that has been collecting n-grams for more than two decades, and its processed corpora are estimated to total more than 7 billion tokens [4]. Therefore, we assumed that using an n-gram collection of such volume could provide better results than expected. Moreover, the main use case in this research was not to build a typical prediction model, but to construct a language graph to better represent the n-gram collection and use it to explore different features.

Natural language processing has been heavily influenced by advancements in the field of machine learning. New paradigms have been successfully applied in the fields of speech recognition and speech synthesis, as well as in machine translation. In addition, new methods have improved the modeling of cognitive processes that are essential for natural language understanding and natural language generation. With novel machine learning techniques, n-gram based language models have been somewhat neglected as the use of neural networks provided promising results. However, not all models and techniques provide equally successful results for all languages. Surprisingly, research has shown that neural network-based language models are less successful for morphologically rich languages than for morphologically poor languages. Moreover, n-gram based language models are significantly cheaper to train in terms of the computational power required [27][28]. Representing language as a graph opens up possibilities for Graph Neural Network approach, which could provide better results in case of predicting specific features [29].

IV. PROPOSED ARCHITECTURE

System architecture based on microservices has proven as the most convenient approach, because it enabled development of a modular system consisting of an arbitrary number of independent services, with each service providing a specific set of functionalities. In the initial research phases, the interaction system was a single service providing recognition of simple, predefined spoken commands and capable of generating simple notifications. This process was almost entirely rule-based, so interaction service was heavily dependent on rule management and rule processing services. Introducing more advanced capabilities regarding spoken interaction has resulted with additional services being added to the system.

Interaction services provide functionalities which are related to interaction with human users (e.g., creation of spoken notifications or recognition of spoken commands). These services rely on other specialized services to perform generic operations which are required in the process (e.g., n-gram search). Finally, the simplest interaction services enable communication with external systems providing functionalities like speech recognition and speech synthesis.

A. Interaction Services

As previously explained, a strong motivation in design of the proposed system was efficient extensibility with various applications, emphasis being on communication between human users and the system using spoken language. Additionally, the system should be able to generate and reproduce spoken notifications. Services described in the following subsections are related to natural language generation tasks.

A.1 Template Management Service

Template management service enables definition of textual templates which are used for generation of notifications. In most cases, the purpose of a spoken notification is to inform the user of a specific state or an event occurring in the environment. Therefore, it is important to reference specific parameters from the environment directly in the template. When the notification is generated, the referenced parameter will be substituted with the current value. Additionally, template management service enables definition of conditions (or "triggers") which will start the notification generation process. This is achieved by creating rules which are evaluated by the rule processor service every time a change in the environment is registered. If the specified rule conditions are met, the notification will be created. Another possibility is scheduling notifications by defining periodic or fixed time generation.

A.2 Notification Generator Service

Notification generator service orchestrates the process of generating a notification from the previously defined template. In this process, it depends on several services. It is invoked by the rule processor service when conditions for generating a notification are met. The process itself relies on data provider service which substitutes the environment parameters referenced in the template with current values. The initially created notification is then corrected in a process employing the n-gram search service and morphological lexicon service. The process is described in detail in the following section.

A.3 N-gram Search Service

N-gram search service provides a query interface towards a graph database constructed from a large n-gram collection. N-

grams are word sequences comprised of N words. Typically, n-gram-based language models are constructed from large text corpora, from which the word sequences are extracted. The constructed model therefore contains information about word transitions as they occurred in the source corpora. The n-gram model described in this paper was constructed by using a n-gram database constructed from a large general vocabulary text corpus collected by Hascheck, a Croatian spell-checker service. Currently, N-gram search service is based on 3-gram system, which contains word sequences comprised of three words. However, this can be changed in future research since Hascheck contains n-gram collections of different lengths, with $2 <= n <= 7$ [4].

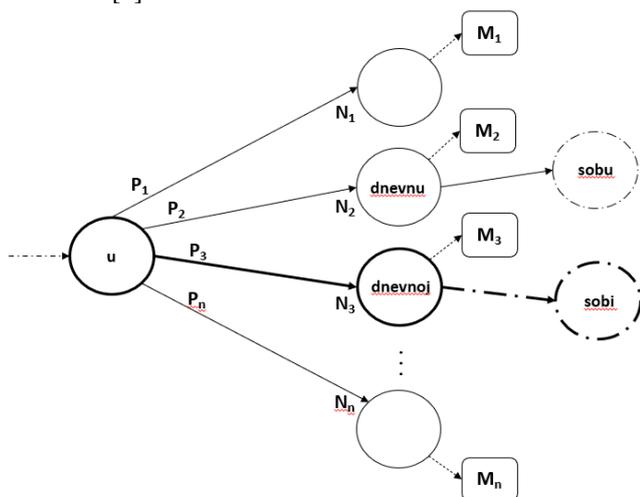
Fig. 1. The n-gram based Croatian language graph

N-gram system used in scope of our work is represented by a directed graph. In this representation, words are represented by nodes, while connections in the graph represent the existence of linguistic connection, with the associated probability. This is depicted in Fig. 1. Analyzing n-gram systems is very useful in case of morphologically rich languages, such as Croatian. They provide information about exact word forms in the given linguistic context. This enabled us the ability to predict the most probable following word based on the given word or a sequence of two words. Additionally, this approach enabled us to determine the probability of words preceding the given word.

Fig. 1. represents an example of several nodes from the language graph. The example is tailored to emphasize the morphological difference in a simple linguistic construct. We can see two similar sequences - "u dnevnoj sobi" and "u dnevnu sobu". In both cases, the corresponding English translation would be "in the living room". However, in Croatian language, these two sequences are used in different contexts – the first is used when referring to a location where something is (for instance – "something is in the living room"), while the second is used when indicating movement (e.g., "I will put something in the living room").

A.4 Morphological Lexicon Service

Morphological lexicon service provides information regarding morphological forms. For a given word, it will provide its morphological features. Additionally, it can provide the correct morphological form for a given word and the specified morphological features. It can also provide the entire morphological tree for the given word.

Croatian is a morphologically rich language, which means that words typically occur in various forms when used in a sentence. For instance, nouns are determined by their gender, number and case; verbs by verb tense, aspect, mood, and voice. Using correct morphological forms is especially important in scope of spoken notifications because it results with more correct and credible generated notifications which sound more naturally when reproduced if the spoken words are in appropriate forms.
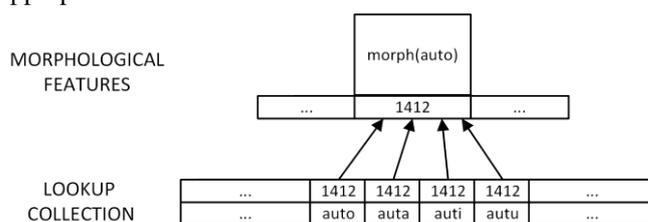

Fig. 2. Morphological lexicon – lookup collection and corresponding morphological data

Morphological lexicon service relies on a dictionary constructed using a collection of words with morphological features available in scope of Hascheck spell checker service. Words are organized in two collections, one being a lookup for the given word and its key which points to related data in the other collection, which contains all morphological variations and their corresponding morphological features. The lookup collection is important because it enables finding the given word in constant time, with search time complexity being $O(1)$. Additionally, the lookup collection solves the ambiguity problem. In Croatian language it is common that the given word does not always change despite being in a different morphological form. Therefore, the lookup will lead to the multiple possible morphological features which could be applicable. The lookup collection and collection containing morphological features are displayed in Fig. 2, with example values. The correct morphological form can only be determined by analysing the linguistic context, which is explained in Section V.

Regarding organization of morphological features, morphological lexicon is constructed similarly to MorphoLex system [30], a morphological lexicon for English language. A similar system exists for Croatian language, as well: Croatian Morphological Lexicon [31]. However, it is a web-application not usable in scope of a real-time application in a smart environment.

In the current implementation, morphological lexicon service cannot determine morphological features if the given word is not present in the dictionary. This could be improved in the future, but would require significant extensions. Since Croatian is morphologically rich language, there are many specific rules which would need to be implemented. However, there are also many exceptions which would need to be explicitly written in a dictionary.

A.5 Text Normalization Service

Text normalization service performs transformation of given text into pronounceable units. This is important for spoken notifications, because text needs to be normalized before being sent to Hascheck Voice to be synthesized. In this process, all non-standard words and symbols need to be identified and replaced with their corresponding pronounceable counterparts. This includes numbers, time and date, acronyms, abbreviations, and symbols (e.g., "8" becomes "eight", "°C" becomes "degree Celsius", etc.). Text normalization in Croatian language is significantly more challenging than in English, because each full extended form needs to be correct in terms of gender, case, and number. However, Text Normalization Service does not perform morphological analysis or correction. In the current implementation, text normalization is performed only if the generated textual notification needs to be synthesized. In that case, the process described in Section V will be applied.
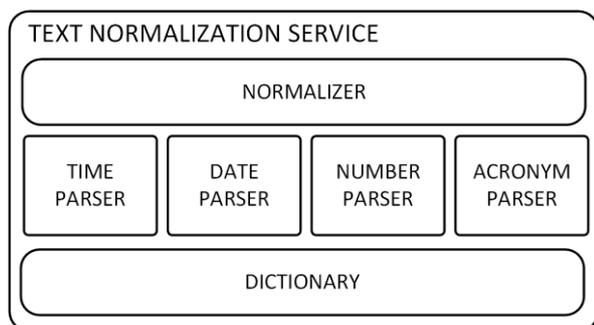

Fig. 3. Text normalization service overview

Text normalization process begins with identification and classification of non-standard tokens in the given notification. Classification of tokens is performed by using regular expressions. Once classified, every non-standard token is then transformed according to its type. Fig. 3 depicts how the service is organized internally. Text normalization service employs a rule-based approach combined with a predefined dictionary. In case of numbers, time and date, specific rules are applied in transformation process. Symbol, acronym and abbreviation normalization is performed by using a dictionary, where normalized forms are contained.

Context is important for text normalization. Depending on the context in which the given content is observed, its normalized form can differ significantly. For instance, common symbols such as "-", ":", ".", or "," should be handled differently in a scientific (e.g. mathematical) context than in standard literary context. Therefore, context-dependent normalization profiles should be introduced, which is a topic for future research.

A.6 Hascheck Voice Connector Service

Connector services in scope of interaction system enable communication with external systems providing capabilities such as speech synthesis or speech recognition. For purposes of reproducing spoken notifications, we used Hascheck Voice, a service providing general vocabulary speech synthesis in Croatian language, developed on Faculty of Electrical Engineering and Computing, Zagreb. It can be used as a service in the cloud, but it can also be deployed on a node in the given smart environment, thus avoiding additional dependency on external systems.

Hascheck Voice Connector Service communicates with the corresponding external system by using HTTP protocol. Hascheck Voice requires normalized text content which should be synthesized and returns the generated audio content.

B. Deployment Options

In a given environment, interaction nodes with custom service sets can be deployed in different physical locations. This enables adjustment to both the target environment and human users. Interaction nodes need to be equipped with appropriate interaction devices and deployed in physical locations adjacent to human users.
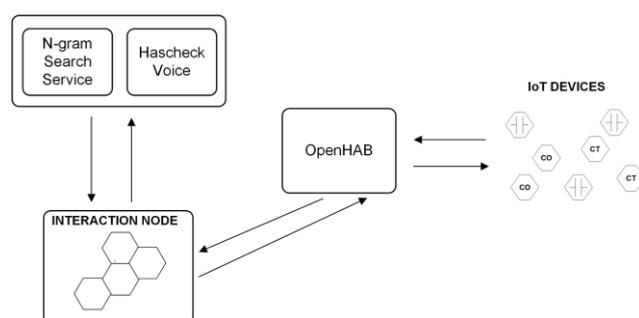

Fig. 4. Integration with OpenHAB IoT platform

Since there are many notable IoT platforms, we decided to explore options for integration of natural language processing services with existing platforms. The motivation was to explore possibilities for providing spoken interaction using Croatian language where it was not natively supported. In this deployment scenario, the existing IoT platform performs monitoring and control of devices in the smart environment. Communication with end-devices is always performed by the IoT platform. Based on information retrieved from the IoT platform, additional metadata can be defined, as well as specific rules. Finally, the interaction system enables capture and interpretation of user requests, and construction of notifications. For this purpose, we decided to explore OpenHAB (Open Home Automation Bus) [32], an open source IoT platform with well-defined application programming interfaces which enabled integration with our custom interaction subsystem. This is shown in Fig. 4. Integration with OpenHAB proved us that it is possible to successfully extend existing IoT platforms with spoken interaction capabilities. Additionally, it has revealed which services are required to build a generic NLP extension.

Another interesting approach is deployment of services on the mobile edge. This could be especially convenient for emergency services and other essential purposes. In this case, the interaction subsystem could perform independently,

offering wide coverage while not being reliant on cloud infrastructure, thus significantly increasing availability. In such deployment scenario, requests would be geographically distributed among the base stations. This could present an additional advantage, since it would enable creation of localized social and environmental context, allowing development of advanced context-aware systems and applications. There is always a concern regarding scalability, but the concept seems promising and opens possibilities for further research.

V. SPOKEN NOTIFICATIONS

Natural language generation can be described as a process of creating natural language in textual form. However, this is a very vague statement, since there are many methods which can be employed in the process, as well as many different purposes. For instance, natural language generation can be employed for generating notifications from data collected in the given environment, but also for dialogue with a user or summarization of a larger amount of information such as news articles. In the current scope of our research, natural language generation is used for creating notifications based on data available in the given IoT environment.

Data collected in smart environments corresponds with devices and their capabilities. However, regardless of devices and their purposes, the data collected is structured and can thus be easily interpreted. Therefore, the method which can be employed in the natural language generation process is template-based. This means that textual notifications are generated using a set of predefined textual templates which reference the actual parameters from the environment. In the process of constructing the notification, the referenced parameters are then swapped with their corresponding values.
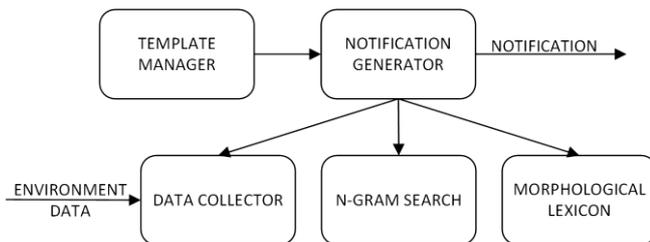

Fig. 5. Services involved in notification generation process

Due to specific features of Croatian language, additional steps are required to make the generated content morphologically correct. This is important because it impacts the quality of user experience, as shown in recent research [22]. Additionally, using the incorrect morphological forms may even influence the meaning of the conveyed information. Using a template-based approach morphological features can be specified directly in the template, but this significantly increases complexity of the templates. Therefore, we decided to explore a different approach, in which the notifications would be generated using simple templates and corrected afterwards. The proposed process for generation of notifications in Croatian language is realized as a set of services, as shown in Fig. 5.

Template manager enables users to define templates for custom notifications and corresponding trigger conditions. When the specified conditions are met, the textual notification is constructed by notification generator service. The notification is based on the previously defined template and available data collected from the environment. In the notification generation process, two generic services are used for determining the correct morphological forms – N-gram search service and morphological lexicon service. Data collector represents a link with external data sources (e.g., IoT platform or a custom set of devices).
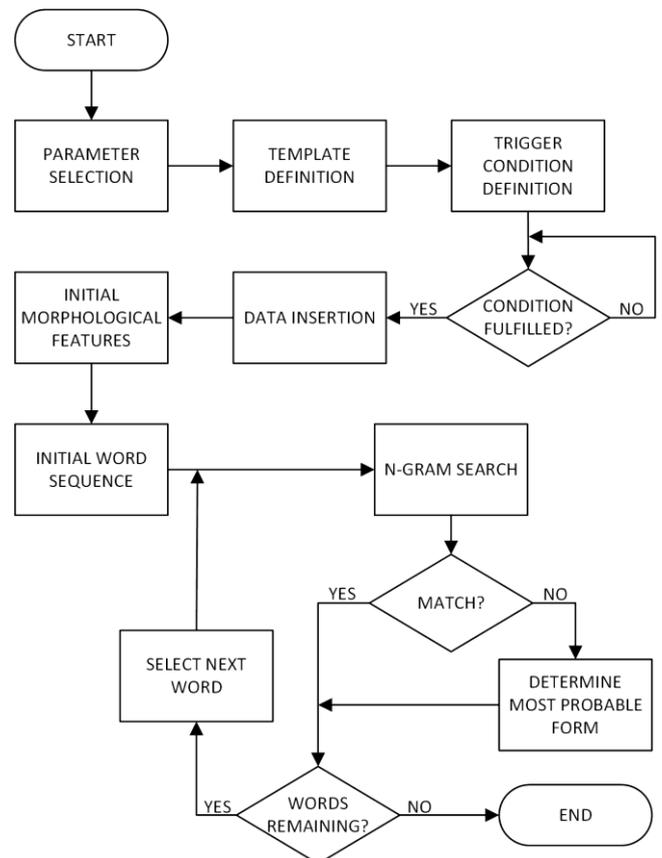

Fig. 6. Notification generation process

The detailed process of notification generation is shown in Fig. 6. The initial step is selecting the parameters from the environment which will be used when constructing the notification. Then a template is created, defining the notification content. Next, the trigger conditions must be defined. Trigger conditions represent specific conditions in the environment which will invoke the process for generating the notification. Once the trigger conditions are defined, the given template is inactive until the specified conditions are fulfilled. When conditions are met, the current values of selected parameters are inserted into the template, resulting with the initial version of the textual notification. For every word in the notification its morphological features are then determined. These features are provided by the Morphological Lexicon Service. The initial word sequence is then selected. Currently, the observed word sequences consist of 3 words. This corresponds with the underlying n-gram system, which is

currently based on 3-grams. N-gram search is performed for the selected sequence, and the results are then analyzed. If the morphological forms from the notification do not match the n-gram search result, the appropriate word forms are determined according to the greatest probability of occurring in the underlying n-gram system. If the morphological forms match, there is no need for corrections. The procedure then continues for each following word in the generated notification.

VI. CASE STUDY – SPEECH-ENABLED APPLICATION IN A SMART HOME ENVIRONMENT

Providing a speech interface to a computer system can be a valuable enrichment in various domains. In a smart home environment, spoken interaction can be employed in various use case scenarios which can positively influence user experience. Therefore, we decided to explore how the described system could be used in this domain. For this purpose, we equipped the IoT laboratory on our Faculty with devices which could be employed in a smart home scenario. Our smart home environment was equipped with a desktop computer and two remote system nodes, each node being a Raspberry Pi computer. On first node the OpenHAB IoT platform was deployed, while the other node hosted a set of previously described services. The desktop computer (Intel i7-6700 CPU, 3.40 GHz, 16 Gb RAM, SSD data storage) was hosting Neo4j graph database containing the n-gram based language graph. Additionally, it hosted the related N-gram search service and Hascheck Voice - the speech synthesis service. All interaction services were implemented as Java microservices using Spring framework.

The OpenHAB node was performing direct monitoring and control of IoT devices in the environment. The reason for such organization is practical – in scope of our solution, connector services should be developed for each device used. Instead, we relied on functionalities provided by OpenHAB which enabled integration with end-devices. Therefore, in scope of our solution, it was required only to use an HTTP client to communicate with OpenHAB API. Additionally, a web-application was developed for users to define custom metadata for end-devices. It enabled users to define custom keywords and descriptions for devices in the environment and their available actions. For this purpose, all information available in scope of OpenHAB was retrieved and extended with user-defined entries. The set of IoT devices used in scope of our simulated smart home environment included presence sensors, microphones, Philips HUE smart lights, temperature and humidity sensors, light sensor, thermostat device and a smart lock. A speaker was connected directly to the interaction node for purpose of reproducing generated spoken notifications.

Spoken notifications were defined using a set of templates related to some of the devices from the environment and their parameters. The purpose was to enable users in the environment to receive updates regarding specific devices with device parameters set dynamically. Examples of notification templates and resulting notifications are given in Table 1.

The feedback in form of spoken notification will be reproduced if there is a template associated with the specific conditions. For instance, a notification template with trigger condition related to smart lock device will generate and reproduce the notification each time the given device parameter value has changed. For instance, a user can change the smart lock parameter value by locking or unlocking the door. In that case the parameter value will change, and a notification will be reproduced informing user of the successful operation. For some notifications it has proven more convenient to be triggered at a specific time or using a periodic time interval. Such notifications represent simple status reports and are convenient in case of volatile environmental parameters such as temperature, humidity, etc.

TABLE I
TEMPLATE AND NOTIFICATION EXAMPLES

ID	Template	Notification
1	Ulazna vrata su <lock1.value>.	Ulazna vrata su zaključana. (*The front door is locked.*)
2	Termostat u <termo2.location> je postavljen na <termo2.value>.	Termostat u kuhinji je postavljen na 21 °C. (*Thermostat in the kitchen is set to 21 °C.*)
3	Temperatura u <temp1.location> je <temp1.value> <temp1.unit>.	Temperatura u dnevnoj sobi je 22 °C. (*Temperature in the living room is 22 °C.*)
4	Ulazna vrata su <lock1.value>. Vanjsko svjetlo je <light1.value>. Temperatura u <temp1.location> je <temp1.value> <temp1.unit>.	Ulazna vrata su zaključana. Vanjsko svjetlo je ugašeno. Temperatura u dnevnoj sobi je 22 °C. (*The front door is locked. Outer light is turned off. Temperature in the living room is 22 °C.*)

Spoken notifications were evaluated in terms of user experience. The emphasis was on how useful notifications were in the given context and how good was the quality of the reproduced notification. When evaluating quality of the reproduced notifications, morphological correctness and quality of the synthesized speech were considered. Preliminary evaluation results show promise, but more extensive testing and detailed analysis are yet to be performed.

Regarding performance, notification generation results are provided in Table 2. For each notification template, the average duration of generating spoken notification is listed. In the process, querying the language graph represents the most time-consuming operation, because it may be repeated multiple times for a single notification. Typically, a single graph query duration varies from 26 to 83 ms. Therefore, we noted how many graph queries were performed in case of each notification.

TABLE II
NOTIFICATION GENERATION DURATION EXAMPLES

ID	Graph queries	Parameters	Total duration (ms)
1	2	1	372
2	6	2	715
3	6	3	826
4	10	5	1193

Performance evaluation has shown that notification generation also depends on number of parameters used in the notification template. This is not surprising, since parameter values are subject to morphological correction. Additionally, the number of used parameters correlates with notification length, which results with more graph queries being performed. The presented experiment shows that the most complex notification can be generated in approximately 1200 ms, which enables real-time communication in a smart environment. However, these results would not be satisfactory in case of a dialogue, since additional processing related to user input and more complex response generation would be required in that case. Therefore, optimization techniques would be required, similar to case described in [33].

In the graph database instance used in this experiment, there were 5104 word nodes and approximately 12000 connections between them. The resulting graph database size was slightly less than 600 Mb. In case of more complex language graphs, the query performance would be reduced. However, in-depth performance analysis related to language graph complexity and comparison between hardware infrastructure used are planned for future research activities. This analysis will be important for the possible deployment scenario in the mobile edge, in which there are limited hardware resources.

Another relevant feature of the proposed approach is the morphological accuracy of the produced notifications. As previously discussed, the performance in this sense depends on the number of words and their morphological variations included in the language network. For the presented approach which is focused on use in a smart home environment, the current graph database contained all the word instances and corresponding relations used in the generated notifications. Therefore, generated notifications were successfully corrected and final results were morphologically accurate. However, in general cases where the graph database would not contain all of the word instances used in the generated notifications, results would not be as accurate if the identical procedure was applied. Therefore, a domain-specific approach when constructing the graph database would be recommended.

VII. Conclusion

In this paper, we presented a model for spoken notifications in smart environments using the Croatian language. The described model relies on an n-gram based Croatian language network, previously presented in [5], for morphological correction of notifications generated from structured data in the smart environment. Notifications are generated based on user-defined templates that contain parameters related to specific measurements from the given environment. These functionalities involve complex processes and therefore rely on standalone services that perform specific tasks from the natural language processing domain. The evaluation was conducted in a simulated smart home environment equipped with a desktop computer, two remote system nodes and a set of smart IoT devices. The described model was found to be functional and achieved satisfactory performance in the smart home scenario. We believe that the described approach could be applicable in other domains, such as industrial environments. However, it is difficult to estimate the different requirements in different application domains, since the performance depends on several factors, such as language graph complexity, the length of the notification, the number of parameters and measurements from the environment, etc. Therefore, additional usage scenarios will be evaluated in future research. For this purpose, additional devices and associated metadata will be introduced. Finally, the interaction model can be extended and improved by enabling human users to issue spoken commands. Additional experiments regarding hardware performance should also be conducted, too. Finally, Graph Neural Network techniques should be thoroughly analysed and applied to the language graph to predict various linguistic features.

References

[1] Tadic, M., Brozovic-Roncevic, D., Kapetanovic, A.: "Hrvatski jezik u digitalnom dobu", META-NET White Paper Series. Springer, Heidelberg etc., 2012.

[2] Vasić, D., Brajković, E.: "Development and Evaluation of Word Embeddings for Morphologically Rich Languages", In 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, pp. 1-5, 2018. DOI: 10.23919/SOFTCOM.2018.8555822

[3] Dembitz, S., Knezevic, P., Sokele, M.: "Hascheck-The Croatian academic spelling checker", In Applications and Innovations in Expert Systems VI, pp. 184-197, 1999. DOI: 10.1007/978-1-4471-0575-6_14

[4] Gledec, G., Šoić, R., Dembitz, Š.: "Dynamic N-Gram System Based on an Online Croatian Spellchecking Service". IEEE Access 7, 149988-149995, 2019. DOI: 10.1109/ACCESS.2019.2947898

[5] Simunec, M., Soic, R., and Vukovic, M.: "N-gram Based Croatian Language Network", 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2021, pp. 1-5, 2021. DOI: 10.23919/SoftCOM52868.2021.9559112

[6] Alexakis, G., Panagiotakis, S., Fragkakis, A., Markakis, E., Vassilakis, K.: "Control of Smart Home Operations Using Natural Language Processing, Voice Recognition and IoT Technologies in a Multi-Tier Architecture", Designs 3(3), 32., 2019. DOI: 10.3390/designs3030032

[7] Afanasev, M.Y., Fedosov, Y.V., Andreev, Y.S., Krylova, A.A., Shorokhov, S.A., Zimenko, K.V. and Kolesnikov, M.V.: "A Concept for Integration of Voice Assistant and Modular Cyber-Physical Production System", In 2019 IEEE 17th International Conference on Industrial Informatics (INDIN) (Vol. 1, pp. 27-32). IEEE, 2019. DOI: 10.1109/INDIN41052.2019.8972015

[8] Zheng, Y., Liu, Y. and Hansen, J.H.: "Navigation-orientated natural spoken language understanding for intelligent vehicle dialogue", In 2017 IEEE Intelligent Vehicles Symposium (IV) (pp. 559-564). IEEE, 2017. DOI: 10.1109/IVS.2017.7995777

[9] Shahrestani, S.: "Internet of things and smart environments". Cham: Springer international, 2018.

[10] Berdasco, A., López, G., Diaz, I., Quesada, L., Guerrero, L.A.: "User Experience Comparison of Intelligent Personal Assistants: Alexa, Google Assistant, Siri and Cortana", In Multidisciplinary Digital Publishing Institute Proceedings, Vol. 31, No. 1, 51-59, 2019. DOI: 10.3390/proceedings2019031051

[11] Ford, M., Palmer, W.: "Alexa, are you listening to me? An analysis of Alexa voice service network traffic", Personal and Ubiquitous Computing, Vol. 23, No. 1, 67-79, 2019. DOI: 10.1007/s00779-018-1174-x

[12] Martinčić-Ipšić, S., Pobar, M., Ipšić, I.: "Croatian large vocabulary automatic speech recognition", Automatika, Vol. 52, No. 2, 147-57, 2011. DOI: 10.1080/00051144.2011.11828413

[13] Bajo, D., Turković, D., Dembitz, Š.: "Rapid prototyping of a croatian large vocabulary continuous speech recognition system", INFOCOMP 2013, p.13, 2013.

[14] Pobar, M.: "Sinteza hrvatskoga govora utemeljena na odabiru jedinica i stohastičkim modelima", Doctoral dissertation, University of Zagreb. Faculty of Electrical Engineering and Computing, 2014.

[15] Meštrović, A., Bernić, L., Pobar, M., Martinčić-Ipšić, S. and Ipšić, I.: "A croatian weather domain spoken dialog system prototype", Journal of computing and information technology, 18(4), pp.309-316, 2010. DOI: 10.2498/cit.1001916

[16] Kocijan, K., Bekavac, B. and Šojat, K.: "Language Generation from DB Query", In Proceedings of the Linguistic Resources for Automatic Natural Language Generation-LiRA@ NLG, pp. 25-32, 2017. DOI: 10.18653/v1/W17-3804

[17] Saratlija, J., Šnajder, J. and Bašić, B.D.: "Unsupervised topic-oriented keyphrase extraction and its application to Croatian", In International Conference on Text, Speech and Dialogue, pp. 340-347, 2011. DOI: 10.1007/978-3-642-23538-2_43

[18] Soic, R., Skocir, P. and Jezic, G.: "Agent-based system for context-aware human-computer interaction", In KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, pp. 34-43, 2019. DOI: 10.1007/978-3-319-92031-3_4

[19] Soic, R., Vukovic, M., Skocir, P. and Jezic, G.: "Context-aware service orchestration in smart environments", In Agents and Multi-agent Systems: Technologies and Applications 2019, pp. 35-45, 2020. DOI: 10.1007/978-981-13-8679-4_3

[20] Soic, R. and Vukovic, M.: "Natural Language Agents in a Smart Environment", In Agents and Multi-Agent Systems: Technologies and Applications 2020, pp. 17-26, 2020. DOI: 10.1007/978-981-15-5764-4_2

[21] Šoić, R., Vuković, M. and Ježić, G.: "Spoken notifications in smart environments using Croatian language", Computer Science and Information Systems, (00), pp.36-36, 2020. DOI: 10.2298/CSIS2004 24036S

[22] Soic, R., Vukovic, M. and Jezic, G.: "Speech Controlled IoT System Based on Context-Driven Rule Engine", In 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1-7, 2020. DOI: 10.1109/INISTA49547.2020.9194642

[23] Gomez, C., Chessa, S., Fleury, A., Roussos, G. and Preuveneers, D.: "Internet of Things for enabling smart environments: A technology-centric perspective", Journal of Ambient Intelligence and Smart Environments, 11(1), pp.23-43, 2019. DOI: 10.3233/AIS-180509

[24] Cohen, P.R., Oviatt, S.L.: "The role of voice input for human-machine communication", In Proceedings of the National Academy of Sciences, 92(22), pp.9921-9927, 1995. DOI: 10.1073/pnas.92.22.9921

[25] Jurafsky, D. and Martin, J. H.: "Speech and Language Processing", 2nd ed., Pearson Prentice Hall, 2008.

[26] Reiter, E. and Dale, R.: "Building Natural Language Generation Systems", Cambridge University Press, 1999. DOI: 10.1017/S1351324 997001502

[27] Cotterell, R., Mielke, S.J., Eisner, J. and Roark, B.: "Are All Languages Equally Hard to Language-Model?", In Proceedings of NAACL-HLT pp. 536-541, 2018. DOI: 10.18653/v1/N18-2085

[28] Shareghi, E., Gerz, D., Vulić, I. and Korhonen, A.: "Show some love to your n-grams: A bit of progress and stronger n-gram language modeling baselines", In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4113-4118, 2019. DOI: 10.18653/v1/N19-1417

[29] Schlichtkrull, M.S., De Cao, N. and Titov, I.: "Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking", In International Conference on Learning Representations, 2020.

[30] Sánchez-Gutiérrez, C.H., Mailhot, H., Deacon, S.H. and Wilson, M.A.: "MorphoLex: A derivational morphological database for 70,000 English words", Behavior research methods, 50(4), pp.1568-1580, 2017. DOI: 10.3758/s13428-017-0981-8

[31] Tadić, M. and Fulgosi, S.: "Building the Croatian morphological lexicon", In Proceedings of the 2003 EACL Workshop on Morphological Processing of Slavic Languages, pp. 41-45, 2003. DOI: 10.3115/1613200.1613206

[32] OpenHAB: https://www.openhab.org/

[33] Tsai, V., Baumann, T., Pecune, F. and Cassell, J.: "Faster responses are better responses: Introducing incrementality into sociable virtual personal assistants", In 9th International Workshop on Spoken Dialogue System Technology, pp. 111-118, 2019. DOI: 10.1007/978-981-13-9443-0_10

Renato Šoić is a research assistant at the Department of Telecommunications of the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He received the master's degree from the University of Zagreb, in 2010. He participated in many industrial projects from different domains, including monitoring and control systems in satellite industry, mobile payment services and large-scale analytics and recommendation systems. Renato Šoić has co-authored seven conference articles and three journal articles. His research interests include speech technologies and human–computer interaction in smart environments.

Marin Vuković is an assistant professor at the Department of Telecommunications of the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He received the Ph.D. from the University of Zagreb in 2011. Marin Vuković has co-authored over 40 journal and conference papers and reviewed a number of papers for international conferences and journals. He actively participated in panels, round tables and held invited lectures with the goal of popularization of science and profession. He is a co-author of the patent at the Croatian Institute for Intellectual Property. Marin Vuković is a deputy director of "Laboratory for Security and Privacy (SPL)" and "Laboratory for Assistive Technology and Alternative and Augmentative Communication (ICT-AAC)" at the University of Zagreb, Faculty of Electrical Engineering and Computing. He is a senior member of IEEE Communications Society.