Arabic Dialect Detection using LLMs

Senior

Progress Report 2

جـــامعـــة الـشـــارقـــة
UNIVERSITY OF SHARJAH

Student Name	UID
EmadEddin Abdulsalam Al-Chmri	*U22105598*
Moheeb Suliman Musa Suliman	*U22105916*
Isa Al-Khanous	*U22200681*
Mohammad Ra'ed Mohammad Hardan	*U22105630*

Computer Science

Supervisor: Dr. Ashraf Mohamed Al-Nagar

College of Computing and Informatics

University of Sharjah

November 2025

Contributions

Student Name	Role	Contributions
EmadEddin Abdulsalam Al-Chmri	Frontend Programmer	Website programming, n8n frontend setup.
Moheeb Suliman Musa Suliman	UI Designer, Frontend Tester	Website UI design, Website integration, analysis, and testing.
Isa Al-Khanous	Backend Programmer	Backend implementation, n8n testing.
Mohammad Ra'ed Mohammad Hardan	n8n Specialist, Backend Tester	n8n hosting, setup, and integration, backend testing.

Abstract

Since our last report, we have implemented a working prototype at last. Using the n8n platform, we achieved an interactive backend that synergizes with the frontend as we grow ever closer to the planned goal of a multi-agent Arabic dialect detection system. However, we have faced a multitude of setbacks that dragged our schedule down by a significant degree.

Nonetheless, we proved our concept's viability and, despite the aforementioned hurdles, the final system will be complete by the deadline with enough hard work and dedication. This requires thorough testing and integration, including revamping our placeholder systems with ones that are both fully functional and sports a fascinating design. In the end, the project will become one we shall be proud of.

Research

Before moving on, we decided to split our four-member group into two: one team worked on the backend, while the other worked on the frontend. This dramatically boosted our progress as we developed the project in parallel while distributing the varied requirements efficiently based on our strengths.

The main focus of our efforts as of this period was n8n, a curious website that hosts the power to bring the multi-agent aspect of our project to life. With a brief overview of its structure, we created a hypothesized chain of operations that have to be deployed in order to link our frontend website with n8n in a cohesive manner:

1. User sends in a message through the telegram or frontend

2. Message will be captured through a webhook or a telegram trigger node

3. The message is forwarded to multiple connected LLMs for dialect detection.

4. The linked models will respond, voting on their classification.

5. Once a decision has been made, a second prompt is given to the highest performer to produce a suitable response to the input in the appropriate dialect.

6. n8n sends the response back to the user on the same platform

This flow is perfect in terms of disseminating the input and retrieving the best output as a reply. Using the n8n's own documentation and with further experimentation, we realized it uses webhooks and telegram triggers to introduce connections between systems, and displayed the operations using node diagrams with flexible input and output flows. This alone was immensely beneficial, but we had no starting point to base our next steps off of. Thus, we utilized the myriad tutorials across the internet about the service, and were educated by our patient Supervisor in the ways of its implementations.

Meanwhile, a frontend had to be designed in order to interact with n8n as intended. We decided on a website-based implementation due to its similarity with the major LLM models available on the internet, its relative simplicity when scaled, and also because of our lack of experience in creating full-fledged mobile applications.

Initially, the website, named Lisan, was non-functional and only displayed basic LLM widgets, but it rapidly showed promise once we added webhooks and a send-recieve pipeline to and from n8n. Though this all provided simple feedback about the working state of the system, it can only be described as a prototype. In addition, the website was made with rudimentary HTML, CSS, and JavaScript, which left a lot to be desired in terms of aesthetic as well, so research was made on ways to enhance it.

Challenges

The most substantial challenge we faced was, without a doubt, time management. Due to the unfortunate amount of workload upon each member in the project, it was quite rare that two or more members would be able to work in tandem. This was exacerbated by the mid-term exams happening in mid-to-late October, halting all progress for a while; the only thing possible to do then was to conduct more research.

Another significant struggle was with n8n itself, as the backend team was surprised by its hidden complexity, affording more time to troubleshooting than to testing. Another option proposed to bypass this was to use a Telegram chat as middleware, a drastic measure that needed to be analyzed thoroughly. This definitely required more focused effort to resolve, as without it, the project cannot advance.

Furthermore, another obstacle for our multi-agent system is the accessibility of other models through n8n, some being more premium than others. The prices of such models would be considered a worthy investment, but only if we can finalize the pipeline with the available free models, such as Gemini.

In terms of the frontend, the current interface underwent a drastic revamp from the base template developed for it. The design was made in reference to our Junior Project video, which was admittedly constructed

without feasibility in mind. Our Supervisor distinctly pushed us toward a more eye-catching design, even if it demanded discarding the current design.

Progress

As described above, we have accomplished the first step in our multi-agent system on both the frontend and backend side. Although the measures we have taken need to be reassessed, our current system works as a proof of concept.

Figure 1.1: The LISAN Website Initial Design

This design was decided upon when the Junior video was made, and it served us well for testing. However, it does not meet our standards for our final vision, so it will change in the future.

Figure 1.2: A Conversation with LISAN

Once a conversation is initiated and a dialect is identified, the background will shift to highlight the detected region as the LLM responds appropriately. Due to the current design's quirks, the chat logs are not as visible as we hoped, something we have to keep in mind going forward when remaking it.

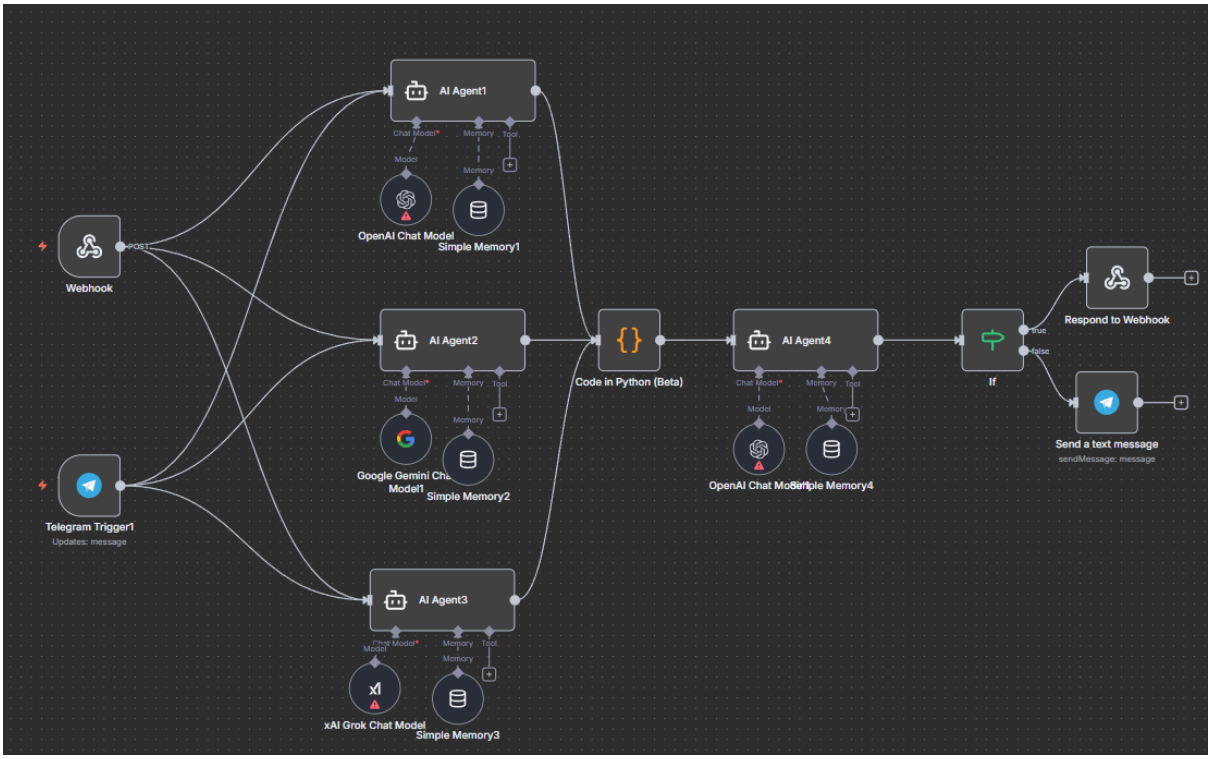

Figure 2: n8n Node Diagram

This figure illustrates the core backend architecture of our system, implemented within the n8n automation platform.

Two Triggers: The process begins with two distinct input mechanisms: a Webhook node, which receives data from our frontend website, and a Telegram Trigger, which captures live user messages from a Telegram bot.

Multi-Agent System: Upon receiving an input, the workflow distributes it in parallel to three distinct AI agents.

This mutli-agent setup is the core of our dialect detection system, allowing multiple LLMs to independently analyze the same input simultaneously. After that, the outputs are run into a python code which will include a weighted voting system. The compiled data is then sent to the

OpenAI model, which processes it to produce the final, contextually appropriate response in the identified dialect. An If node then routes the response to the appropriate channel: either to the frontend via "Respond to Webhook" or to the Telegram user.

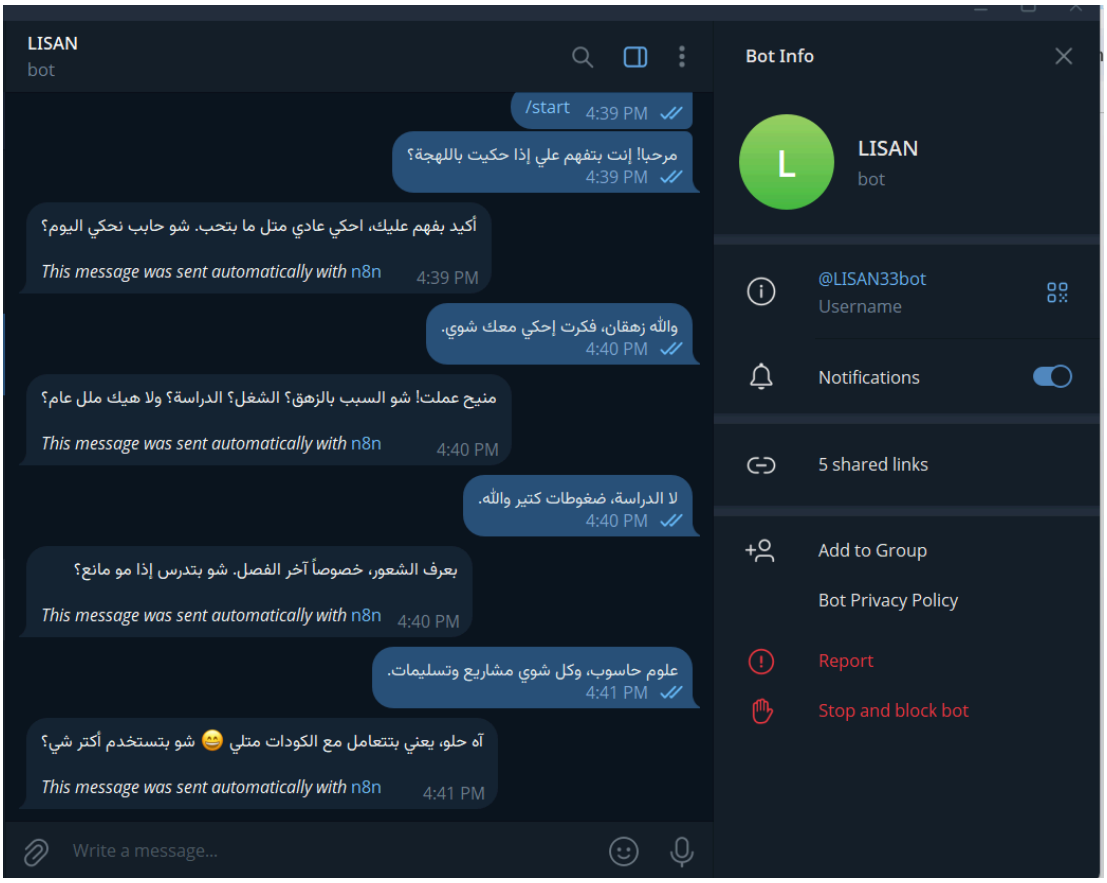

Figure 3: The LISAN Telegram Bot

This figure demonstrates the end-to-end functionality of our n8n workflow using Telegram as a live frontend through a real conversation with the LISAN bot (@LISAN33bot).

Reflections

The current phase of the project has been a valuable learning opportunity in both the technical side and the project management. We witnessed the shift from theoretical planning to actually implementing something, and we were able to create a working prototype that demonstrated the concept we had conceived. The decision to break into a frontend team and a backend team was a good decision and allowed us to work on things simultaneously while utilizing everyone's strengths.

On the technical side, we were able to work with a new tool: n8n. We learned how to do node-based workflow logic, and thought about the nature of webhooks in the overall integration of the system. We also learned a valuable lesson about the unexpected challenges of new tools and how that can complicate the development time. Time management proved to be one of the challenges of the project, particularly based on our academic obligations outside of the project. It did teach us a possible lesson learned with regard to project proposals, which is potentially allowing time for buffering into the associated timelines.

Certainly we did learn to communicate about any time management concerns, indicating that we were the best intent on things going smoothly. All in all, even with challenges, our commitment to getting a central basic "communication" method developed between the two sides was a confidence booster and created potential for our work.

Milestones Achieved

1- Team Structure and Parallel Development:

Successfully divided the team into specialized frontend and backend units, significantly accelerating the development process.

2- n8n BackEnd Prototype:

Established a functional n8n workflow that can receive input, distribute it to LLM agents, and handle a basic voting mechanism for dialect classification.

3- FrontEnd Website ("LISAN"):

Developed a basic but operational website interface using HTML, CSS, and JavaScript that can send user input to and receive responses from the n8n backend.

4- Comprehensive Research:

Identified and evaluated a wide range of potential LLMs (e.g., Falcon, Jais, ALLaM) suitable for Arabic dialect detection and response generation.

5- Proof-of-Concept Pipeline:

Integrated the frontend and backend to create a working prototype, demonstrating the end-to-end flow from user input to system response, thus proving the project's core viability.

Future Milestones

1- LLM Integration and Multi-Agent Enhancement:

We plan to expand our multi-agent framework to include three more advanced LLMs (DeepSeek, Grok, and ChatGPT) to increase the diversity and robustness of our dialect classification system.

2- Frontend Design Overhaul:

As expressed above, we desire the frontend to be something eye-catching and distinct from standard LLMs in terms of design, animations, content, and so on.

3- Full-Scale Hosting:

As it is, the website is hosted locally. In the future, if further expansion becomes our goal, we shall determine a suitable platform or service to host our complete system for public use.

4-End-to-End System Testing & Refinement:

Conduct thorough integration testing to ensure stability, reliability, and speed of the entire system.

5-Documentation and deployment:

Compile comprehensive technical and user documentation, and finalize all deliverables for project submission and demonstration.