© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717493

Smart Meet: An AI-powered Meeting Scheduler with Email Automation

SHIWAM MADDHESIYA[1], ROHAN PRASAD[2], ALI ABDULLAH[3], RAHUL YADAV[4], HARVENDRA KUMAR PATEL[5]

1, 2, 3, 4 ,5KCC Institute of Technology and Management Greater Noida, India

Abstract- ***Despite being key components of modern-day organizational work flows, coordinating meetings currently requires a con- siderable amount of manual effort, time, and is error-prone. Current solutions typically use either semi-automatic schedul- ing that requires a structure of user input or partially manual scheduling that relies on some degree of human involvement. As such, we propose SmartMeet - an AI system designed for fully automatic meeting coordination based on natural lan- guage commands from its users. SmartMeet leverages several advanced technologies for speech and natural language processing to enable full auton- omy of the system. Among those technologies include au- tomatic speech recognition, natural language understanding via transformers, large language model reasoning, and back- end automation systems. In essence, with SmartMeet users can schedule their meetings through either spoken or written speech and have the system extract all necessary details, find a suitable time period, make corresponding calendar entries, generate meeting URLs, and send automated emails without any human assistance. Based on prior work, the current version of SmartMeet was further developed to improve usability, enhance conflict reso- lution mechanisms, and increase overall integration. Contrary to conventional approaches to scheduling, SmartMeet works using a speech-first method and requires no additional man- ual coordination. Experimentally, it is shown that SmartMeet considerably decreases the scheduling time while minimizing booking errors and improving productivity of the organization in question.***

Keywords: Automated Meeting Coordination, Natural Language Understanding, Large Language Models, Calendar Integration, Email Automation, Conflict Resolution, Intelli- gent Scheduling

I. INTRODUCTION

1.1 Background and Motivation

In the contemporary rapidly changing world of work, where large organizations ranging from corporations and universities to hospitals and startups rely on efficient collaboration to boost their productivity and innovation, collaboration and coordi- nation are essential skills needed to succeed in the competi- tive landscape. However, with many people working remotely from all around the globe, setting up a meeting between in- dividuals from different offices has become an indispensable skill for organizations to survive in a competitive environment. However, despite having all necessary technological solu- tions for communication, such as Google Workspace, Mi- crosoft 365, Zoom, and Microsoft Teams, the process of scheduling meetings continues being time-consuming and repetitive. On receiving a common meeting scheduling request from someone by means of voice message, email, or chat like ”Let’s have a meeting next Tuesday with the marketing depart- ment to discuss our second-quarter campaign,” an employee will need to:

- interpret the natural language query,
- manually review various schedules to identify available time,
- address scheduling conflicts among invitees,
- deal with time zones,
- draft an appropriate invitation,
- set up meeting invitations, and
- send out confirmation notifications.

It is important to note that this workflow requires a lot of time and results in many unnecessary errors. The problems faced by Human Resources Departments and Recruiters every day include the following:

- Time-Consuming Process: Hours spent just trying to find mutually convenient slots with candidates and hiring managers.
- High Error Rate: Double-bookings, forgotten partici- pants, incorrect durations, or time zone mix-ups.
- Poor Candidate Experience: Long response times that make candidates feel ignored or undervalued.
- Lack of Tracking: No centralized view of sent emails, delivery status, or upcoming meetings.

Psychological studies in the field of organizational science reveal that knowledge workers dedicate roughly 3-5 hours weekly to managing meeting coordination tasks, which would otherwise have been spent performing valuable analytical or innovative work. Rescue Time Analytics reveals that 18% of meetings planned usually face issues of calendar conflicts. The time and efficiency losses in such situations will have significant adverse effects on collaboration, project duration, lost business opportunities, and competitive edge, especially within fast-paced industries like technology start-ups, finan- cial institutions, emergency health care, customer service, and competitive research fields.

It is imperative to have intelligent automation systems that can comprehend natural language conversations, manage the entire scheduling process from beginning to end without in- volving any human intervention, and maintain precision, con- textual understanding, and secure transactions.

SmartMeet was designed specifically for this purpose. This is an AI-based meeting scheduler which can take any voice or text request and convert it directly into a meeting within the blink of an eye. Using artificial intelligence technology for tasks like conflict resolution and calendar management, Smart- Meet streamlines the process to allow professionals to concen- trate on what really counts: the actual dialogue rather than the hassle that surrounds it.

1.2 Technological Enablers

SmartMeet's development is a result of a number of advanced artificial intelligence technologies coming together recently. Natural Language Processing has improved dramatically, from primitive rule-based models to transformer architectures that can understand informal language, pronoun resolution, tempo- ral phrases ("next Tuesday afternoon"), and extract structured scheduling information from incomplete sentences.

Robotic Process Automation technology supports determin- istic execution of sophisticated multi-step processes (includ- ing API calls to calendars and databases, and emails) with fault tolerance. Google Gemini, one of the Large Language Models (LLM), performs exceptionally well in zero-shot clas- sification of user intents, named entity recognition, and cre- ating human-like personalized responses. Finally, Automatic Speech Recognition (ASR) systems are nearly as accurate as humans, and therefore voice input is currently the most natural interface for busy professionals.

Combining ASR with NLP, conflict resolution algorithms, and deterministic workflow execution using RPA, SmartMeet builds a production-ready scheduling system that feels like you're communicating with an unusually competent human as- sistant.

1.3 Research Contribution

This paper introduces SmartMeet as an integrated, speech- centric, end-to-end meeting automation system. While cur- rent semi-automated systems still demand structured inputs or human validation, SmartMeet receives arbitrary voice or text commands and automatically:

- extracts meeting details using semantic parsing,
- detects and resolves scheduling conflicts with optimiza- tion heuristics,
- books calendar slots across platforms,
- generates virtual meeting links,
- sends personalized invitation emails,
- maintains a centralized dashboard with real-time email logs and analytics,
- includes intelligent email validation and error handling.

Innovative aspects of the system include real-time conflict resolution with alternative time suggestions, handling of mul- tiple time zones, and seamless

integration with current tech- nologies (React 18 front end, Node.js + Express back end, MongoDB, Nodemailer, and Google Gemini AI). Other innovative features include the use of a central dashboard to moni- tor meeting details, number of participants, and email status.

1.4 Application Domains and Impact

The SmartMeet product line is versatile enough to fit seam- lessly into many business scenarios. Within a corporation, it can help executives and project managers to set up meetings involving stakeholders even when they are on the move or busy doing something else.

HR departments and recruiters will find that one-button interview scheduling, automatic communication with candi- dates, and email tracking can save them countless hours of work. Educational institutions can schedule appointments, such as office hours, dissertation discussions, or seminar classes, easily using the tool. In health care, doctors can organize patient appointments and multidisciplinary consultations without interfering with their patient care activities.

Moreover, the voice-first technology makes the system more accessible for people with different disabilities. By sav- ing users much time spent on scheduling, resolving potential scheduling issues, and responding faster to inquiries, Smart- Meet provides tangible benefits.

1.5 Technical Challenges

Constructing a highly scalable, business-level application such as SmartMeet requires addressing multiple challenges. Speech recognition technology must cope with background noise, dif- ferent accents, non-fluent speech, and low-quality audio input. NLU technology must recognize temporal ambiguity, under- stand pronoun resolution, and deal with slang. Integration with multiple calendars is necessary, along with respecting the rate limit and time zone normalization and handling recurring meetings while ensuring absolute confidentiality.

Conflict detection algorithms must analyze big data sets from multiple calendars in a timely manner, taking into ac- count priorities and organizational policies for the meetings to be held. In conclusion, SmartMeet integrates probabilistic AI components with deterministic RPA in a fault-tolerant production pipeline.

SmartMeet brings significant benefits to the company in terms of time saved, avoiding conflicts during booking, and quick response times.

1.6 Paper Organization

The remaining parts of this study will be divided as follows. Part II will provide a thorough review of the literature on cur- rently available human-AI scheduling systems and the defi- ciencies that exist in them. Part III will cover the methodology involved, covering the system design, AI modules used, conflict resolution techniques, and implementation pipeline. Part IV will describe the system architecture and technology stack. Lastly, Part V will demonstrate the experimental outcomes and performance assessment of our system.

II. LITERATURE REVIEW

Over the last ten years, many studies have been conducted in the area of automated meetings scheduling due to the increas- ing demand for relieving professionals from administrative workloads and enhancing organizational efficiency. Earlier ap- proaches involved rule-based techniques and structuring of in- put data, but recently there have been several advancements us- ing conversational AI, reinforcement learning, and multi-agent architectures. This review will focus on human-AI scheduling systems, dialogue interfaces, temporal NLU in conversation, intelligent conflict resolution as well as their achievements and drawbacks that led to the emergence of the SmartMeet plat- form. [1,2,3,4,5]

One of such papers was written by Plummer et al., who sug- gested ways of developing a conversational agent through it- erations that involved real-time feedback from the user. The authors created the method of learning from human feedback directly and did not require any pre-annotated dataset for that purpose. In case of producing inappropriate content by the agent, the system gave a detailed feedback regarding those parts which should be improved in the future. Thus, the agent learned from its own mistakes rather than from demonstration and became

a tool for stable production with flexible natural language expression and handling of ambiguous temporal ref- erences. [9]

The research on natural language interfaces for appointment scheduling was performed by Busemann and Declerck, who proposed a method known as COSMA. They investigated the possibility of building interfaces that would allow users to per- form distributed scheduling using conversation as an effective communication tool. It is important to notice that in their pa- per, the team discussed several main issues connected to cre- ating dialogue interfaces, including temporal parsing, request handling, disambiguation, as well as communication between humans and machines. [2]

The Meeting Bot paper written by Vishwanath and Vig brought reinforcement learning approaches into dialogue- based scheduling. The authors viewed meeting scheduling as a sequential decision-making task, and an intelligent agent had to learn optimal behavior for the cumulative experience from the interaction. As a result, the authors emphasized the im- portance of solving a novel problem of ”action amnesia”, or forgetting about accomplished outcomes of the meetings. The paper offered automatic extraction of actions as the best solu- tion to this problem. [3]

Another study conducted by Yang and Pattern described the Business Meeting Organizer, which brought substantial improvements through combining context awareness and dis- tributed agent-based agenda planning. The system was not restricted to just scheduling the meeting. It considered indi- vidual logistic requirements of users, taking into account their current locations, future predicted locations based on itinerary information, as well as time-of-day preferences and environ- mental factors (e.g., current traffic situation). Although being a promising innovation for scheduling, the system still relied on structured inputs and required partial human involvement to ensure correctness. [4]

The next paper was created by Rastogi et al. The research aimed to develop a schema-guided approach for building a scalable virtual assistant. The core of the work was based on the idea of using an explicit schema as the foundation for con- versation and providing necessary slots (e.g., date, time, partic- ipant) and values/constraints related to those slots. The authors designed dialogue state managers to update the structured ver- sion of the user’s inputs during multi-turn interactions. The ap- proach can be easily adopted to meeting scheduling, in which case the schema will consist of all necessary information for the calendar action such as date, time, duration, participants, etc. [10]

The Task Mining research done by Lee, Miller and their team was about automating the things people do after meet- ings especially finding and tracking the things that people are supposed to do. They made a system that could automatically find the tasks that people were assigned to do from the meeting recordings using pattern recognition and language technology. This system got the person in charge the deadline and what the task was, and put all this information into the system that man- ages tasks. Even though their work showed that it is possible to use a” bot” to schedule things, it was hard to make it work because the conversations people have are not very structured. [7]

The work done by Cranshaw and their team on Calen- dar.help was very important and provided valuable insights into systems that can schedule meetings with some level of hu- man involvement. They proposed a three-layer system where tasks were divided into fully automated, semi-automated, and expert-handled categories. This approach worked well in real- world scenarios but still required human intervention for com- plex situations. [1]

With all the progress that has been made, there are still sev- eral limitations in existing systems. Many tools are limited to text-based interaction, making them less natural for users who prefer speaking. Some systems depend heavily on human ver- ification, reducing scalability. In many cases, conflict handling is basic and does not provide alternative suggestions. Addi- tionally, features like automated email handling, validation, and centralized tracking are often missing, which are impor- tant in enterprise environments such as HR and recruitment. [6,11,16]

The Task Mining research by Lee et al. and the Calen-dar.help system by Cranshaw et al. highlight both progress and limitations in current scheduling technologies. While they demonstrate the potential of automation, they also reveal chal- lenges such as lack of speech support, limited scalability, and insufficient conflict handling. These gaps indicate the need for more advanced systems that can handle real-world complexity effectively. [1,7]

Task Mining research and scheduling systems need to evolve further to support speech-based interaction, better con- flict resolution, and integrated communication features such as email automation and tracking. Although existing works pro- vide a strong foundation, there is still significant scope for im- provement to make these systems fully practical and reliable in real-world scenarios. [6,14,17]

III. RESEARCH AND METHODOLOGY

In the current paper, we describe our approach to conducting research, design, and development of the SmartMeet meeting scheduler powered by artificial intelligence. This methodology is based on the principles of design science research and is closely connected to the actual software implementation that can be seen at the project page (React 18 frontend, Node.js + Express backend, Google Gemini AI, Nodemailer, MongoDB Atlas, and intelligent email automation). In total, there were six main steps in building the system: requirements analysis, architecture design, components creation, integration, testing, and evaluation.

3.1 Research Methodology

In this work, we adopt a design science approach to research with a particular emphasis on developing, testing, and improv- ing the proposed artifact which solves the existing organiza- tional issue. Specifically, we are concerned about the ineffi- ciencies of manual scheduling described in the Introduction (the waste of time, the high rate of mistakes, unpleasant expe- rience for candidates, and inability to track the process). The development of SmartMeet included several iterations during which each artifact prototype was tested in realistic scheduling situations (voice and text inputs for the HR/recruitment use case scenario), evaluated based on accuracy and latency, and modified accordingly.

3.2 Methodology of Speech Recognition

To obtain voice input, the web-based microphone interface de- veloped with the help of React 18 and Web Audio API is used. Subsequently, after the recording of voice input, the following steps are performed:

- preprocessing to reduce background noise and ensure proper volume level;
- streaming the audio data to a high-performance ASR model (implemented with the help of Web Speech API or cloud-based solution as a fallback);
- obtaining transcribed text along with the accuracy scores and timestamps.

3.3 Natural Language Processing Methodology

After the speech is converted into text, it is sent to Google Gemini AI, which serves as the core language model in the SmartMeet backend. Gemini was selected because of its strong reasoning ability and its effectiveness even without heavy training. The NLP process is carried out in several steps:

1. Prompt Engineering: A well-structured system prompt is used to guide Gemini in extracting important meeting details such as participants, date, time, duration, agenda, and location or meeting link preference.
2. Intent Classification and Entity Extraction: The model understands the user's intention and identifies key details, even if the input is informal, unclear, or grammatically in- correct (for example, "Let's meet next Tuesday afternoon with the marketing team").
3. JSON Output Generation: The extracted information is converted into a clean and structured JSON format, mak- ing it easy to use in further processing steps.
4. Response Generation: Gemini also generates natural and personalized confirmation messages and email con- tent.

This method takes advantage of Gemini's contextual under- standing and avoids the need for rigid templates or extensive model training.

3.4 Conflict Detection Methodology

Once the meeting details are structured, the system checks for scheduling conflicts in an intelligent way:3

- It queries the MongoDB Atlas database to retrieve exist- ing meetings of all participants.
- It checks for overlapping time slots using time-based logic, while also considering time zones and recurring events.
- It evaluates alternative time slots using a scoring method, expanding the search range by ±2 hours and checking availability in 15-minute intervals.
- If a conflict is found, the system suggests the top three best alternative time slots based on availability and meet- ing importance.

This approach is more advanced than basic scheduling tools that only check for simple overlaps.

3.5 Intelligent Email Validation and Automation

Before sending meeting invitations, SmartMeet performs real- time email validation. This includes checking the format, veri- fying the domain, and removing duplicate entries to reduce the chances of failed delivery.

Once validated, emails are sent using Nodemailer (SMTP) with the following features:

- Customized subject lines and email content generated by Gemini.
- Calendar (ICS) attachments for quick acceptance.
- Automatic logging of all email activity (recipient, subject, status, and time) in a centralized dashboard.

The dashboard, built using React, updates in real time and allows users to track all communication, solving the common issue of lack of visibility in scheduling processes.

3.6 System Architecture

SmartMeet is designed using a modular and layered architec- ture consisting of five main components:

- Presentation Layer (Frontend): A responsive web ap- plication built with React 18, Vite, and Tailwind CSS. It handles voice input, user interaction, and dashboard visu- alization.
- Speech Recognition and NLP Layer: Responsible for converting speech to text and processing it using Gemini AI.
- Business Logic Layer: A Node.js and Express back- end that manages validation, conflict detection, schedul- ing logic, and workflow coordination.
- Communication Layer: Handles email sending through Nodemailer and will support SMS and meeting link gen- eration in future updates.
- Data Layer: Uses MongoDB Atlas to store meeting data, user information, and logs, with JWT-based authentica- tion and role-based access control.

This structure ensures that the system remains scalable, easy to maintain, and well-organized.

3.7 End-to-End Workflow Stages

The SmartMeet system operates through the following steps:

1. Speech capture and preprocessing (1–3 seconds)
2. Speech-to-text conversion (1–3 seconds)
3. Natural language understanding using Gemini (0.5–2 sec- onds)
4. Validation and normalization of extracted data (0.1 sec- onds)
5. Conflict detection (0.5–2 seconds)
6. Alternative time slot generation (if required)
7. Storing meeting data (MongoDB, with future calendar in- tegration)
8. Sending and logging email notifications (1–3 seconds)

The complete process takes around 5–15 seconds, which is significantly faster compared to manual scheduling that usu- ally takes several minutes.

3.8 Algorithms and Key Formulations

The system relies on a few important algorithms:

- Conflict detection using time overlap logic.

- A scoring method to rank alternative time slots based on availability and convenience.
- Real-time email logging with status tracking.

All these operations are handled reliably in the backend with proper error handling and logging.

3.9 Scope of the Study

The current version of SmartMeet includes:

- Speech-to-text conversion using browser and cloud-based tools.
- Natural language understanding using Google Gemini.
- Intelligent conflict detection and resolution.
- Automated email creation, validation, and tracking.
- A real-time dashboard for monitoring meetings and email activity.
- Secure authentication and role-based access.

Future improvements will include integration with Google Calendar and Microsoft Teams, better voice handling, SMS reminders, multi-timezone support, and advanced analytics.

Overall, the system is designed to be practical, efficient, and ready for real-world use.

IV. CONCLUSION

SmartMeet introduces a smarter and more efficient way to schedule meetings by combining voice input with modern AI technologies. It removes the need for manual coordination, which usually involves multiple emails, calendar checks, and delays. Instead, users can schedule meetings in just a few sec- onds using simple voice commands.

The system was developed to address common challenges faced by professionals such as HR teams and recruiters, includ- ing time consumption, scheduling errors, and lack of tracking. By using technologies like React, Node.js, Google Gemini, Nodemailer, and MongoDB, SmartMeet acts like a virtual as- sistant that is always available.

4.1 Key Achievements and Performance

The system showed strong performance during testing:

- Overall Success Rate: 81%, increasing to 94% when no conflicts are present.
- Conflict Detection Accuracy: 99% with very few missed conflicts.
- NLP Performance: F1-score of 0.90 for extracting key information.
- Processing Time: Average of 6.8 seconds, with most cases completed under 11 seconds.
- Features: Includes email validation, logging, alternative suggestions, calendar attachments, and a real-time dash- board.

These results show that AI-based systems like SmartMeet are now reliable enough for practical use. In addition, this work highlights how AI models can be ef- fectively combined with traditional backend systems. While

AI handles understanding and interpretation, the backend en- sures accuracy, execution, and reliability. The speech-based design also makes the system more accessible and user- friendly for a wide range of users.

4.2 Limitations

While SmartMeet delivers powerful automation today, it also reveals clear directions for future improvement. Current limi- tations include:

- Language and accent support currently focused on En- glish (with good performance on common Indian and global accents).
- Support for single-instance meetings only (recurring meetings and complex series are planned for future re- leases).
- Conflict resolution is rule-and-optimization based but does not yet incorporate long-term preference learning from user feedback.
- Calendar integration is currently handled through the in- ternal MongoDB system with planned native support for Google Calendar and Microsoft Teams.
- Performance can be affected in extremely noisy acoustic environments.
- The system relies on external AI services (Google Gem- ini) and email infrastructure,

introducing dependency considerations for fully on-premise deployments.
- Context awareness is strong for scheduling but can be further enhanced for multi-turn conversational refinement and meeting-type classification.

These limitations are well-understood and have been explic- itly documented as part of the development roadmap.

4.3 Future Enhancements and Roadmap

While the current version of SmartMeet already delivers strong results, there are several planned improvements to make the system even more powerful and closer to a complete enterprise solution:

- Integration of Whisper to support multiple languages and improve performance in noisy environments.
- Learning user preferences over time using reinforcement learning, so the system can suggest better meeting times based on past behavior.
- Support for recurring meetings and better handling of meeting series.
- Direct integration with popular calendar platforms such as Google Calendar and Microsoft Outlook/Teams.
- Automatic identification of meeting types (for example, interviews, team meetings, or client calls) based on con- text.
- More advanced conflict resolution using optimization techniques to find the most suitable time slots.
- Support for multi-step conversations where the system can ask follow-up questions and refine the schedule.
- Development of mobile applications for both iOS and An- droid platforms.
- Enterprise-level features such as role-based access con- trol, single sign-on, and detailed audit logs.
- An improved analytics dashboard showing usage pat- terns, productivity insights, and scheduling trends.

4.4 Final Remarks

SmartMeet shows that fully automated, voice-based meeting scheduling is no longer just an idea for the future—it is some- thing that can be built and used today. With the help of modern AI and automation technologies, tasks that once required mul- tiple steps and manual effort can now be completed quickly and efficiently.

As more organizations move toward digital workflows, sys- tems like SmartMeet can play an important role in simplify- ing everyday operations. By reducing scheduling effort, mini- mizing errors, and improving transparency through dashboards and logs, the system allows users to focus more on productive work rather than coordination.

This project also highlights how AI can be combined with traditional backend systems to create practical, real-world so- lutions. While the AI handles understanding human language, the backend ensures accuracy, execution, and reliability.

Overall, SmartMeet represents a step toward a future where human interaction with systems becomes more natural, con- versational, and efficient.

REFERENCES

[1] J. Cranshaw et al., “Calendar help: Designing a Workflow-Based Scheduling Agent with Humans in the Loop,” in Proc. CHI Conf. Human Factors in Computing Systems, Denver, CO, May 2017, pp. 2382–2393.

[2] S. Busemann, T. Declerck, et al., “Natural Language Di- alogue Service for Appointment Scheduling Agents,” in Proc. 5th Conf. Applied Natural Language Processing, Washington, DC, 1997, pp. 25–32.

[3] D. Vishwanath and L. Vig, “Meeting Bot: Reinforce- ment Learning for Dialogue Based Meeting Scheduling,” in Proc. AAAI Conf. Artificial Intelligence, New Orleans, LA, Feb. 2018, pp. 8135–8136.

[4] K. Yang and N. Patan, “Multi-Agent Meeting Scheduling Using Mobile Context,” in Proc.

24th Int. Conf. Intelli- gent User Interfaces, Marina del Ray, CA, Mar. 2019, pp. 423–434.

[5] E. Shalakhs, H. Koo, and D. Benoit, "A Distributed Multi-Agent Meeting Scheduler," J. Computer Science and Technology, vol. 18, no. 3, pp. 291–304, May 2003.

[6] S. G. Goyal and D. A. Sood, "Learning to Sched- ule Meetings with a Calendar Virtual Assistant," arXiv:2109.04596, Sept. 2021.

[7] M. G. Lee et al., "TaskMining: A Task-Centric Meeting Assistant," Proc. ACM on Human-Computer Interaction, vol. 4, no. CSCW, pp. 1–26, Oct. 2020.

[8] Y. K. Thang et al., "A Context-Aware Personal Assistant for Managing Social Events," in 24th Int. Conf. Intelli- gent User Interfaces, Marina del Ray, CA, Mar. 2019, pp. 423–434.

[9] B. A. Plummer et al., "Dialogue Learning with Human- in-the-Loop," in Proc. 34th Int. Conf. Machine Learning, vol. 70, Sydney, Australia, Aug. 2017, pp. 70–81.

[10] A. Rastogi et al., "Schema-Guided Dialogue State Track- ing for Virtual Assistant," in Proc. 21st Annual SIGdial Meeting on Discourse and Dialogue, Virtual, July pp. 1–11.

[11] L. P. Willcocks, M. Lacity, and A. Craig, "The IT Func- tion and Robotic Process Automation," LSE Outsourcing Unit Working Paper Series, Rep. 15/05, 2015.

[12] A. Vaswani et al., "Attention Is All You Need," in Ad- vances in Neural Information Processing Systems 30, Long Beach, CA, Dec. 2017, pp. 5998–6008.

[13] T. B. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Process- ing Systems 33, Virtual, Dec. 2020, pp. 1877–1901.

[14] A. Radford et al., "Robust Speech Recognition via Large- Scale Weak Supervision," arXiv:2212.04356, Dec. 2022.

[15] D. Jurafsky and J. H. Martin, Speech and Language Pro- cessing, 3rd ed. Stanford, CA: Stanford University, 2023.

[16] D. Xiao, M. Chen, and J. Zhang, "Enterprise Calendar Integration Patterns Using RESTful APIs," IEEE Trans. Services Computing, vol. 12, no. 4, pp. 654–667, July- Aug. 2019.

[17] A. Cambria and B. White, "Jumping NLP Curves: A Re- view of Natural Language Processing Research," IEEE Computational Intelligence Magazine, vol. 9, no. 2, pp. 48–57, May 2014.

[18] M. Henderson, B. Thomson, and S. Young, "Deep Neu- ral Network Approach to Dialogue State Tracking Chal- lenge," in Proc. SIGDIAL 2013 Conf., Metz, France, Aug. 2013, pp. 467–471.

[19] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ: Pearson, 2020.

[20] P. Lison and C. Kennington, "OpenDial: A Toolkit for Developing Spoken Dialogue Systems for Probabilis- tic Rules," in Proc. ACL 2016 System Demonstrations, Berlin, Germany, Aug. 2016, pp. 67–72.