Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2021

João Emanuel da Costa Teixeira

Processamento de fala e linguagem para auxiliar na coordenação de reuniões

Speech and Language Processing to Assist Meetings Coordination

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2021

João Emanuel da Costa Teixeira

Processamento de fala e linguagem para auxiliar na coordenação de reuniões

Speech and Language Processing to Assist Meetings Coordination

"Whatever you do in this life,
it's not legendary, unless your
friends are there to see it."

— Barney Stinson

João Emanuel da Costa Teixeira

Speech and Language Processing to Assist Meetings Coordination

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia de Computadorese Telemática, realizada sob a orientação científica do Doutor Nuno Filipe Correia Almeida, Investigador do Departamento Electrónica Telecomunicações e Informática da Universidade de Aveiro, e do Doutor António Joaquim da Silva Teixeira, Professor Associado com Agregação do Departamento Electrónica Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president

ABC
Professor xxx da Universidade de Aveiro

vogais / examiners committee

José Casimiro Pereira
Professor Adjunto do Instituto Politécnico de Tomar

Nuno Filipe Correia Almeida
Investigador da Universidade (orientador)

agradecimentos / acknowledgements

É com muito gosto que aproveito esta oportunidade para agradecer a todos os que me ajudaram durante estes anos.

Um enorme obrigado aos meus orientadores. Não foi um trabalho fácil, mas no fim valeu a pena. Foi sem dúvida uma excelente experiência que irei guardar comigo para sempre.

Aos meus pais pelo esforço que sempre fizeram para que isto se tornasse realidade.

Aos meus amigos pela ajuda e força que sempre deram. Um abraço em especial aqueles que fizeram parte da minha vida nestes últimos 6 anos em Aveiro, foram sem dúvida os melhores anos da minha vida.

Resumo

As reuniões são uma parte importante do nosso dia a dia. Formais ou informais, presenciais ou remotas, são algo inevitável na nossa sociedade. Nas empresas assumem uma importância ainda maior, sendo decisivas para a definição do seu presente e futuro. Apesar de ser uma área de extrema importância, não foi ainda realizada investigação suficiente para compreender e melhorar a qualidade das reuniões. As tecnologias atuais podem melhorar a nossa compreensão das reuniões, fornecendo dados com maior precisão e/ou que simplesmente não eram possíveis antes. Esta dissertação propõe uma plataforma que pode ajudar a coordenar uma reunião em tempo real, fornecendo informações relevantes para o coordenador e todos os participantes. Para desenvolver o sistema, uma abordagem centrada no utilizador foi adotada, começando com a identificação dos utilizadores-alvo e o conjunto de requisitos derivados dos cenários de uso. O sistema desenvolvido adotou ainda uma arquitetura desacoplada e uma semantic knowledge base para fornecer flexibilidade para futuras evoluções. A prova de conceito integra vários módulos de processamento capazes de converter fala em texto e realizar análise da voz. Um conjunto de reuniões pré-gravadas foi usado para testar o sistema. O sistema apresentado mostrou já ser capaz de fornecer aos coordenadores de reuniões informações úteis e interessantes. Pode extrair um conjunto de estatísticas e apresentá-las na forma de gráficos ou texto. Estes estão disponíveis numa dashboard ou através de alertas. O trabalho apresentado é um primeiro passo e uma primeira prova de conceito. O trabalho futuro é rico e cobre distintas linhas de investigação.

Abstract

Meetings are an important part of our daily lives. Formal or informal, in-person or remote, they are something unavoidable in our society. In companies they assume even greater importance, being decisive for the definition of their present and future. Despite being an extremely important area, not enough research has been carried out to understand and improve the quality of meetings. Current technologies can enhance understanding of the meeting, by providing data with greater precision and/or that was simply not possible before. This dissertation proposes a platform that can help coordinate a meeting in real-time, providing relevant information for the coordinator and all participants. To develop a proof-of-concept system, a user-centered Design approach was adopted, starting with the identification of target users and the set of main requirements derived from usage scenarios. The developed system adopted a decoupled architecture and a semantic knowledge base to provide flexibility for future evolutions. The proof-of-concept integrates several processing modules capable of converting speech to text and doing voice analysis. A set of existing pre-recorded meetings was used to test it. The presented system showed to be already capable of providing meeting managers with useful and interesting information. It can extract a set of statistics and present them in the form of charts or text. These are available through a dashboard or an alert module. The presented work is both a first step and an initial proof-of-concept, the future work is rich and covers distinct lines of research.

Contents

Contents i

List of Figures v

List of Tables vii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Problem(s) . . . 2
- 1.3 Objectives . . . 2
- 1.4 Structure . . . 3

2 Background and Related Work 4
- 2.1 Meetings . . . 4
  - 2.1.1 The problem . . . 5
  - 2.1.2 Making meetings work . . . 5
  - 2.1.3 Online Meetings . . . 6
- 2.2 Recent related work in meeting assistance . . . 6
  - 2.2.1 Cognitive Assistant that Learns and Organizes (CALO) . . . 7
  - 2.2.2 Meeting assistant at Intel . . . 8
  - 2.2.3 The Meeting Project at ICSI . . . 9
  - 2.2.4 TalkTraces . . . 9
  - 2.2.5 ProMETheus . . . 11
  - 2.2.6 Mr.Jones . . . 12
  - 2.2.7 Analysis of related work . . . 14

3 Scenarios and Requirements 16
- 3.1 Personas and Goals . . . 16
  - 3.1.1 Pedro, Manager . . . 17
  - 3.1.2 Joana, Front-End Developer . . . 18
  - 3.1.3 Francisco, Team leader . . . 19
- 3.2 Scenario(s) . . . 20
  - 3.2.1 Scenario 1 . . . 20
  - 3.2.2 Scenario 2 . . . 20
  - 3.2.3 Scenario 3 . . . 20
  - 3.2.4 Scenario 4 . . . 21
  - 3.2.5 Scenario 5 . . . 21

3.2.6 Scenario 6 . . . 21
3.3 Requirements . . . 22

4 Tools and Technologies 23
4.1 Meeting Platform . . . 24
4.2 Speech and Language Processing . . . 26
4.2.1 Speech-to-Text . . . 26
4.2.2 Voice analysis . . . 26
4.3 Database / Knowledge base . . . 27
4.4 Development of client-server solutions . . . 28
4.4.1 Server-side . . . 28
4.4.2 Client-side . . . 29
4.5 Communication between modules . . . 29
4.5.1 WebSocket . . . 30

5 Proof-of-concept System 32
5.1 General Architecture . . . 32
5.2 Implementation . . . 33
5.2.1 Jitsi Audio Recorder . . . 34
5.2.2 Audio Repository and Streamer . . . 35
5.2.2.1 Voice analysis . . . 35
5.2.2.2 Transcription . . . 35
5.2.2.3 Statistics . . . 36
5.2.2.4 Alerts . . . 36
5.2.3 Knowledge base . . . 38
5.2.3.1 Ontology . . . 38
5.2.3.2 Triple Store . . . 39
5.2.3.3 SPARQL Queries . . . 39
5.2.4 Auxiliary . . . 39
5.2.4.1 Authentication . . . 39
5.2.4.2 Meeting Manager . . . 39
5.2.4.3 Processing of Pre-recorded meetings . . . 40
5.2.5 DashBoard . . . 40

6 Results 43
6.1 Meeting statistics . . . 44
6.1.1 Individual speaking time through the duration of the meeting . . . 44
6.1.2 Intervention Time vs Speaking Time . . . 45
6.1.3 Multidimensional Analysis of the Meeting . . . 47
6.1.4 Participants Mood . . . 49
6.1.5 Transcription . . . 49
6.1.6 PDF generation . . . 50
6.2 Multi-meeting statistics . . . 51
6.2.1 Global Averages . . . 51
6.2.2 User statistics . . . 53
6.3 Alerts System . . . 55
6.3.1 Transcription confidence . . . 56

6.3.2 Speaking Duration Limit 56
6.3.3 Speech Duration departing from average 57

7 Conclusions **61**
7.1 Work summary 61
7.2 Main results 62
7.3 Future work 62

Bibliography **63**

List of Figures

2.1 The CALO-MA conceptual framework (from: [59]) . . . 7
2.2 CALO-MA offline meeting browser (from: [59]) . . . 8
2.3 Meeting assistant at Intel interface (from[2]) . . . 9
2.4 Setup used by the system for presential meetings (from [9]) . . . 10
2.5 The data pipeline for each iteration (from [9]) . . . 10
2.6 ProMETheus system architecture (from [47]) . . . 11
2.7 ProMETheus data pipeline (from [47]) . . . 11
2.8 Mr.Jones architecture (from [8]) . . . 13
2.9 Mr.Jones interface (from [8]) . . . 13

4.1 Tools and technologies used throughout the development of the system . . . 23
4.2 Jitsi Architecture (from [17]) . . . 25
4.3 WebRTC [22] . . . 25
4.4 Triple [52] . . . 27
4.5 Kafka overview . . . 30
4.6 Example of a connection using WebSocket . . . 31

5.1 Conceptual Architecture . . . 32
5.2 Detailed System Architecture . . . 34
5.3 Quick access menu for the meeting coordinator . . . 36
5.4 Group of alert toasts performed by the meeting coordinator . . . 37
5.5 Alert Configuration Interface . . . 37
5.6 Domain Ontology developed, showing classes their relations and properties . . . 38
5.7 New meeting creation . . . 40
5.8 Active meeting tab . . . 41
5.9 Meetings history . . . 41
5.10 Statistics page . . . 41
5.11 Meeting page . . . 42

6.1 Panel from the dashboard presenting the individual speaking time during one meeting . . . 45
6.2 Intervention Time vs Speaking Time . . . 45
6.3 Multidimensional Analysis Chart . . . 47
6.4 Panel from the dashboard presenting the information of mood of speech and total intervention and speech time . . . 49

6.5 Meeting Transcription on the dashboard. From the left to the right it displays the name of the user, date/time, the recognized text and a button to play the original audio. 50
6.6 Transcription PDF file 51
6.7 Results for the first step of the query 6.3 53
6.8 Results for the second step of the query 6.3 53
6.9 Results for the third and last step of the query 6.3 53
6.10 Query result with user values by meeting 54
6.11 Query result for user average values 54
6.12 Chart presenting global average vs individual values 55
6.13 Alert toast showing that the meeting average transcription confidence is low . 56
6.14 Alert toast for participants that exceed the intervention time limit 57
6.15 Alert toasts for substantial more/less intervention time than average. 57
6.16 Group of query results produces from the Query 6.3.3 59

List of Tables

2.1 Related work comparison - Part 1 14
2.2 Related work comparison - Part 2 14
2.3 Related work comparison - Part 3 14

3.1 System Requirements 22

6.1 Meetings processed alongside duration, participants, and number of generated triples 43

Chapter 1

Introduction

> A meeting is indispensable when you don't want to get anything done.
>
> *(Kayser 1990) [44]*

In this chapter, an introduction to the subject of this dissertation is presented. It starts by providing the motivation behind its execution, followed by a description of several problems associated with it. Deriving from these identified problems, a set of objectives is defined that will guide the development of this work. Lastly, a brief explanation of the document structure is presented.

1.1 Motivation

Meetings are a key part of today's society as collaboration is essential. Every day millions of meetings are held around the world [44] and they can be diversified in their nature. For example, they can be formal or informal and face-to-face or remote. Furthermore, meetings are used to address a variety of different subjects from different areas, and possibly make decisions on those subjects.

In the organization life, most companies have implemented regular team meetings [43], as employees have resources and information that could be key to solve their challenges [44]. Meetings also play a large role in employee socialization. Besides the subject in discussion, it's a place where employees get to know each other and build relationships, therefore meetings are very important as they can shape both the team and the organization's outcome. The diversity in topics and the interconnections between participants bring a huge complexity to any research done in this field.

In 2020 the world faced a pandemic and with it, both working from home and remote meetings became standard. Using Zoom as an example, statistics show that they support 300 million participants each day and their meeting minutes have increased by 3300% [27]. For the first time, physical meetings are not the standard, and although they share some similarities they are not the same. Any research and technology developed in this field should have this new conjuncture in high consideration.

Despite the problem being known, the lack of research in this field is evident. Little time is spent searching for ways to improve and help such a challenging process [57]. Being

a challenging and impactful field, technology could be helpful. Providing the right tools to better understand and coordinate a group meeting could facilitate communication and drastically improve productivity.

1.2 Problem(s)

Although meetings are important in our lives, too many are considered a waste of time. According to statistics, 42% of employees classified the meetings they participate as poor and a source of frustration [43]. This number is alarming, especially from an organization's point of view, where no meeting is free. They take time to prepare, schedule and perform, and should bring the solutions needed instead of more problems.

One key element in this problem is the coordinator, someone in charge of mediating the meeting. From managing several members and their turn to speak, to provide clues to facilitate communication. This procedure is not easy, as mediating a meeting among several members is a truly challenging process.

Although it looks like we do not know enough about meetings, it is also clear that not much has been done to change this. A lack of research in this field shows that not enough time is spent searching for ways to improve and help such a challenging process.

Technology can and should help in this regard. There are already several tools from online meeting platforms to transcription services, but making them work together and produce quality data requires too much effort. A system aimed to help in meeting development should consider this. Not only it should produce data easy to interpret by the user, but it should also be built in a way where add/replace modules is an effortless job.

1.3 Objectives

Improving how meetings are conducted is important to achieve better results and can have a positive impact on the performance of the participants. Technology has the potential to be very helpful, giving the right tools to better understand and coordinate a group meeting, improving communication and productivity.

The main objective of this work is to create tools that provide a more detailed view of a meeting, giving important information to better understand and improve its quality.

From this more general objective more concrete ones were drawn:

- Record and analyze the dialog between users.
- The data is both produced and analyzed in real-time, before being stored and shared with everyone in an easy-to-access visualization.
- The system should not be a passive agent inside the meeting. It should advise and alert participants of important information that would improve the development of the meeting.
- As explained before, because of the covid-19 pandemic, meetings changed from physical to remote. The system should not be restricted in this regard, it should work in both situations and with similar outcomes and performance.
- The system should be built in a way where add/replace modules is an effortless job.

Providing the coordinator and participants with relevant information to guide its progression can be a step towards improving employee performance and enthusiasm while helping the organization achieve better results.

1.4 Structure

This dissertation is divided into seven chapters.

Chapter 2 provides some background and related work done in meeting processing, analyses, and coordination. Starting by giving an inside look into what a meeting is, followed by analyzing the most known projects done in the field and ending with a comparison of each other.

Chapter 3 presents the target use case of the system, using a User-Centered Design [23] approach. Starting by identifying the target users and developing Personas, followed by Usage Scenarios that enabled the derivation of a set of requirements.

Chapter 4 introduces the tools used through the development of the system, alongside an explanation for their selection.

Chapter 5 presents the proof-of-concept system, showing the adopted architecture and explaining in detail all the main components.

Chapter 6 presents the results obtained by processing a set of meetings.

In the last chapter, conclusions are drawn alongside some ideas for future work that may evolve or profit from this one.

Chapter 2

Background and Related Work

> Most people doubt online meetings can work but they somehow overlook that most in-person meetings don't work either.
>
> *(Scott Berkun)*

In this chapter, an overview is made on meetings and the work done in this area. It starts by giving a better and deeper inside look at the meeting subject. Throughout this section will be presented statistical information regarding employees and their experience on meetings. This is important to understand how meetings work, their problems, and how it is possible to make them better. In the second part, a set of work done is this subject is explored. They mainly focus on providing tools to better understand and improve meetings. This work is further analyzed and the main features are extracted. The systems are lastly compared to each other using those features.

2.1 Meetings

The definition of meeting is simple, a meeting is when two or more people come together to discuss one or more topics. This discussion leads to a consensus between the participants. If this cannot be accomplished, further meetings must take place, where new viewpoints and ideas will be presented.

Meetings can happen in different ways, from informal to formal, from in-person to remote. It is a major part of today's society as they provide a mechanism that allows people to bring their ideas together, discuss plans, strategies and where decisions with high impact are made [57].

In today's organizational life meetings are essential [57], employees have resources and information that could be key to solve their challenges [43]. They also play a large role in employee socialization as besides the topic in discussion participants also get to know each other and build relationships. According to statistics collected in 2019 [5] employees had, on average, 8 meetings per week, each ranging from 30 minutes to an hour, meaning that they spent around 10% of their work hours in meetings, and this number keeps rising. Although the quantity as increased, the quality seems to be dropping as almost half of the employees

complained that meetings wasted their time. They also felt that the number of meetings is usually excessive.

2.1.1 The problem

Meetings have an important role in today's society but although they are so valuable, a lot of them are seen as a source of confusion and frustration [43]. Its ineffectiveness can have negative consequences such as occupying valuable time to do the actual work or leaving the employees with doubts about the tasks that must be performed. Bad meetings can have an impact, not just internally, but also on the external relationships that the company builds with clients [5].

It is clear that the number of meetings and their length are key factors for their ineffectiveness. Also, the outside work done to prepare the meeting should be taken into consideration as, on average, an employee spends an hour preparing for each meeting and up to half an hour searching for a collaborative space for them to take place [5].

Most meetings have delays that should be analyzed. On average, every meeting has a 10-minute delay and although it doesn't seem much, in the long term, employees can spend days or even weeks' worth of time just waiting for meetings to start.

In organizational life, time is money, not only time is being lost in poorly organized meetings but also no meeting is free for the company. It is estimated that around half a trillion dollars are wasted [5] every year in poorly organized meetings. With the increasing number of meetings, this number is expected to grow.

Meeting attendance is a big problem too as 57% of employees think that improperly attended meetings represent the biggest blow to their company. Shockingly, 73% have admitted to multi-task and 39% have slept during a meeting [5].

Meetings are clearly not working the way they should and despite being such an important field, it has little to no academic research, making it difficult to understand what is a good or a bad meeting [57].

2.1.2 Making meetings work

Although meetings have all the problems above explained they can't be abolished, but they can be improved even if it's so slightly. One of the most important things is that although companies are less hierarchical it does not mean that meetings should not have structure or that their participants do not have different roles. One key element in this problem is the coordinator, someone that is in charge of creating an agenda of topics to discuss and facilitate communication between the participants.

The book "A arte de comunicar com sucesso" [7] is a recommended read and it presents some guidelines to have a more productive meeting:

- Limit attendance, as explained before, attendance can be a big problem. The meeting should be restricted to those that will benefit from it and can have a direct impact on the topics addressed;
- Definition and fulfillment of agenda, as 72% of employees, believe that setting clear objectives is what makes a meeting successful [5];
- Every participant should take part in the meeting, as their opinion matter;

- The coordinator should abstain himself from dominating the meeting;
- Controlling those who intervene too much;
- Preparation of minutes at the end of the meeting for further analysis if needed.

Whether it is coordinating a meeting or participating in one, everyone has an important role to play and duties to fulfill. These rules and obligations work together to produce meetings where things get done.

2.1.3 Online Meetings

In 2020 the world was hit by a pandemic and although work from home was being implemented in some organizations, this conjecture forced everyone to adapt. Companies and organizations started looking for platforms and technologies that can be helpful to support remote meetings. According to Zoom's statistics, their meeting minutes have increased by 3300% in 2021 and the number of daily participants is now around 300 million [28].

Although meetings have a lot of problems, as explained before, the usage of an online platform has shown some interesting results. As it seems losing the face-to-face connection actually has a good impact on employee relationships and engagement. According to some studies, 76% of employees now use video collaboration to work remotely and an astonishing 75% of those experienced increased productivity and an enhanced work-life balance [49]. Most of the respondents to those studies stated that online meetings make it easier to get their point across, it reduces time to complete projects or tasks and helps them feel connected.

These results should be taken into consideration for the future, as it shows that it is possible to have good meetings in remote environments. They should not replace meetings that require an in-person interaction, as they should exist, just less frequently.

Further work and research in this area should have this conjecture in high priority, as it might be the way we communicate in the future.

2.2 Recent related work in meeting assistance

Despite an evident lack of research in this field, some companies and institutions developed tools and technologies that in some way provide a better understanding of meetings and tried to improve their quality. Throughout this section, the most relevant work is addressed. The search was made using the "Google Scholar" engine and the queries presented below. From the analysis of the related work, some key features were extracted. They are used to better compare the systems to each other and to the proof-of-concept system that will be presented further in this dissertation.

Search Engine
Google Scholar

Queries
Meeting Assistant
Meeting technology
Meeting
Meeting coordinator

2.2.1 Cognitive Assistant that Learns and Organizes (CALO)

CALO [61, 58, 59, 39] was a massive collaboration project led by SRI International, funded under the PAL (Personal Assistant that Learns) program of the DARPA. PAL focused on creating a framework for the use of cognitive systems that work alongside humans, learning and applying logic. It worked closely with military partners and it is still now being used to enable voice-based interaction with users to support military planning.

In 2003 the PAL program started the creation of CALO. A five years effort that involved more than 300 researchers with the main goal to facilitate autonomous learning.

As part of the CALO project, the system CALO-MA (CALO Meeting Assistant) was designed to assist meeting participants. A client software records all the participant's audio as well as handwriting recorded by digital pens. All interactions are logged into a database, and after the meeting, further automatic annotation and interpretation technologies are initiated. All the information is available via a web-based interface (Figure 2.2).

Figure 2.1 describes the conceptual framework used by CALO-MA. The speech utterances are delivered to the server which performs real-time and offline processing, the utterance is first recognized and segmented into sentences. Then the sentences are assigned to dialog act tags, which are used to improve action items and decision extraction. Finally, the meeting is segmented into topically coherent segments and summarized according to parameters set by the user.

CALO-MA could retrieve relevant information like commitments and remember dates, projects, places, and people. CALO was also a cognitive system. This means that CALO could extend and improve its capabilities by learning and adapting.

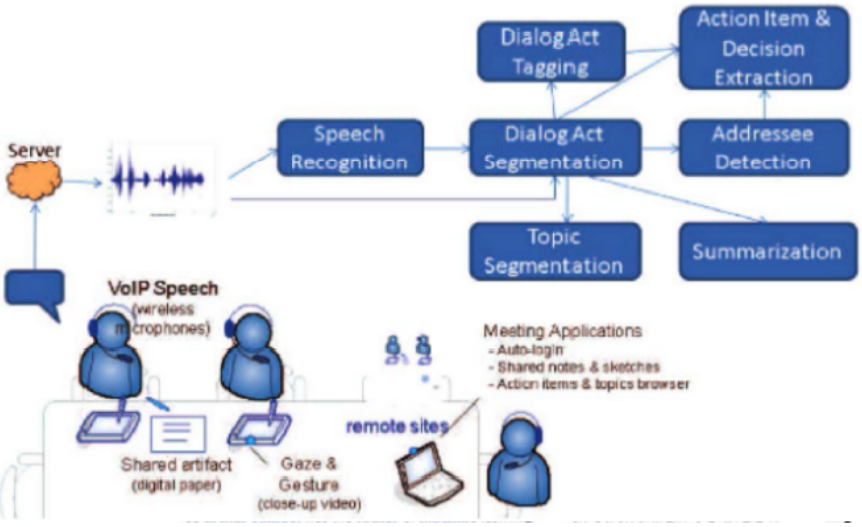


Figure 2.1: The CALO-MA conceptual framework (from: [59])

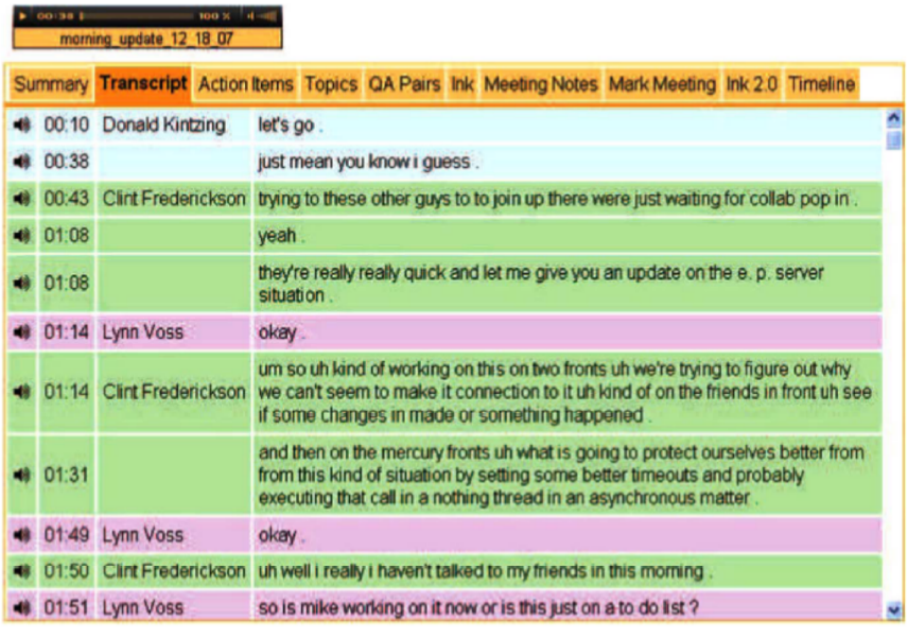


Figure 2.2: CALO-MA offline meeting browser (from: [59])

2.2.2 Meeting assistant at Intel

Meeting assistant [2] is an application developed by eight researchers in 2015, a collaborative job between Intel Corporation Israel, Intel Labs USA, and Intel Corporation Germany.

The system uses four speech and NPL technologies:

- **Automatic Speech Detection (ASR)** using a Deep Neural Network (DNN) for the meeting transcription. Prepared for the 200,000 most frequent words that occur in language model training data and, at this point, no meeting-specific data was used;
- **Speaker Diarization**, the system uses a single microphone without any source of separation or preprocessing. audio is split into 2 seconds segments, each one of them produces a score for all participants. The person corresponding to the highest score is assumed to be the active one;
- **Keyphrase Extraction**, a lightweight, single term and multi-term extraction algorithm that performs sentence splitting, part of speech tagging, and noun phrase chunking to collect keyphrases as potential significant terms.
- **Sentiment Detection**, an NLP application that aims to identify attitudes and opinions from textual documents, since it is highly sensitive to the topical domain, the system used a semi-supervised method to achieve high precision.

Figure 2.3 shows the users interface of the application. The Meeting Assistant records all the audio producing a transcript that is displayed along with NLP metadata. Keyphrases and sentiments are marked - red for negative and green for positive, and the summary based on the keyphrases is also available by pressing the button “Summarize”. On the right side, a word cloud displays the most frequently extracted keyphrases.

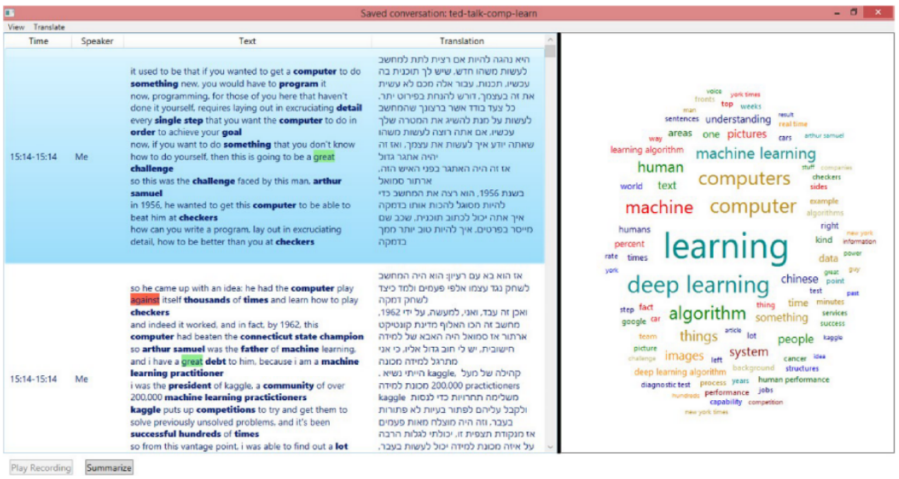


Figure 2.3: Meeting assistant at Intel interface (from[2])

2.2.3 The Meeting Project at ICSI

The Meeting Project was developed by ICSI and funded under the DARPA Communicator project in a subcontract from the University of Washington in 2001. It was focused on transcription, query, search, and structural representation of audio from informal meetings.

The system uses a combination of separate microphones for each participant in addition to 6 far-field microphones and had support to as many as 15 open channels to record the audio, as for the transcription of the meeting two different ways were implemented.

The first one was based on human transcription. A "Transcriber" interface was created where the responsible person would write the transcription of the audio alongside speaker identification and some additional information.

The second one used automatic transcription composed of two modules:

- Recognition system for the transcription used a language model containing about 30,000 words and was trained on a combination of Switchboard, CallHome English, and Broadcast News data, but was not tuned for or augmented by meeting data.
- Segmentation algorithm used partition individual channel recordings into segments of speech. These segments were determined either by automatic segmentation followed by hand-correction, or by hand-correction alone. For the automatic segmentation was used a speech/nonspeech detector.

2.2.4 TalkTraces

TalkTraces [9] was developed by the University of California in collaboration with the Arizona State University in 2019. It is a real-time representation of a conversation with a thematic view of discussion to help teams get a perception of the agenda items that have been covered.

The system works for both presential, as shown in Figure 2.4, and pre-recorded meetings, having two main modules:

- Speech-to-text, using the Google Speech API for real-time conversion;

- Latent Dirichlet allocation (LDA) as a statistical method to identify discussion topics.

Its development was highly focused on user experience and by the end of development two iterations were presented. Figure 2.5 presents the data pipeline for both iterations.

- Iteration 1: features a topic-focused visualization that updates in real-time as the discussion is transcribed and processed using the topic model;
- Iteration 2: additionally to what was achieved in the prior iteration, uses word embeddings to compute agenda-to-discussion similarity in real-time and displays the result to help participants keep track of agenda items covered.

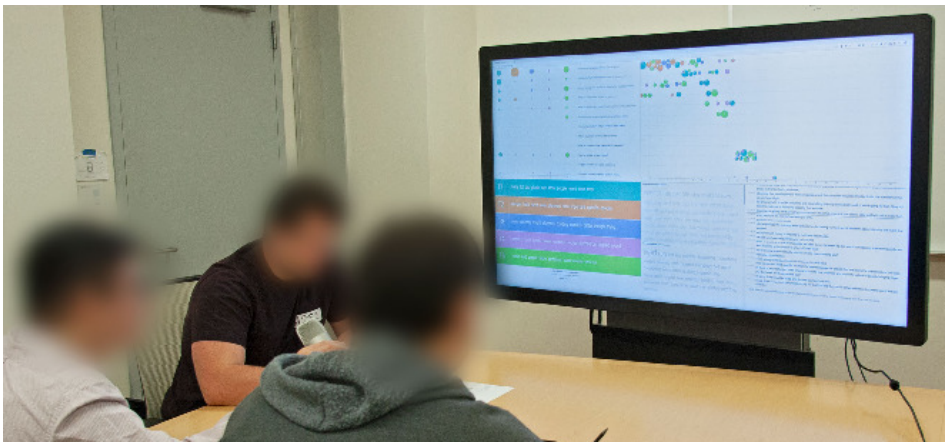


Figure 2.4: Setup used by the system for presential meetings (from [9])

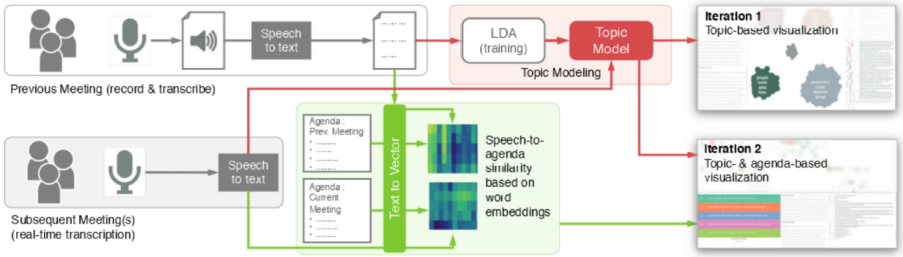


Figure 2.5: The data pipeline for each iteration (from [9])

2.2.5 ProMETheus

ProMETheus was developed by Xidian University in collaboration with RMIT University in 2018. It is focused on generating meeting minutes from audio data. To achieve this, the system has two main modules, Phone Module, and Server Module, as shown in Figure 2.6.

The Phone Module is a mobile application that records meeting audio, uploads the audio data to the Server Module, receives and shows the meeting minutes, as shown in Figure 2.7.

The Server Module processes the audio and generates the meeting minutes. To achieve this, the system splits the audio data into small segments according to the voice activity detection (VAD). After this, all audio pieces would be forwarded to the speaker recognition module to classify different speakers and to the speech recognition module to transcribe the audio into text. This text is further processed using a summarization algorithm that can extract meaningful key phrases, summary sentences, and sentiment analysis, calculating the relevance score of each course by the sentiment and attitude. In the end, the module returns the meeting minutes that consist in meeting contents, summarization, and the agreed action. Everything is done in real-time using machine learning and deep learning.

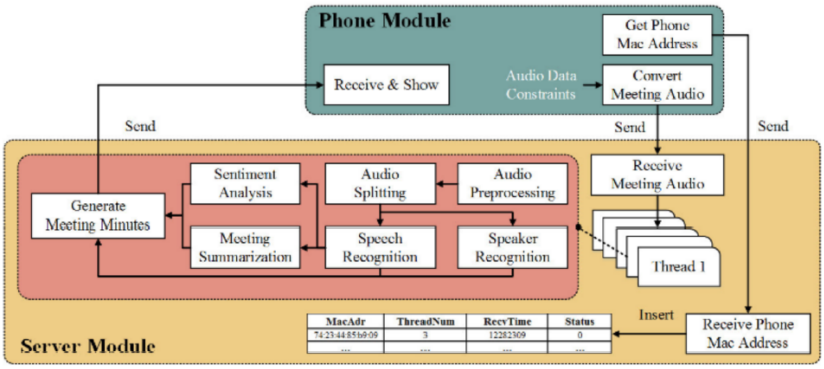


Figure 2.6: ProMETheus system architecture (from [47])

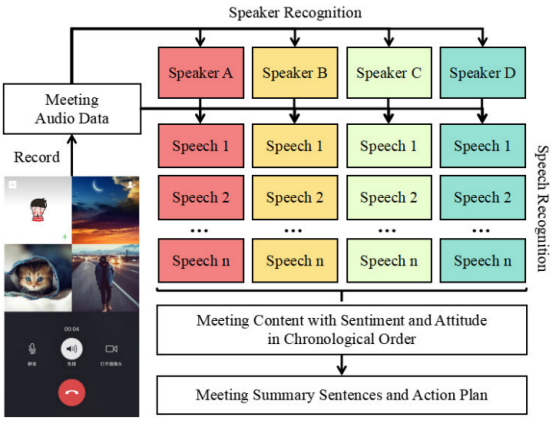


Figure 2.7: ProMETheus data pipeline (from [47])

2.2.6 Mr.Jones

Mr.Jones [8] was developed by IBM and the Arizona State University in 2018 and it is situated at CEL (Cognitive Environments Laboratory), which is a multi-agent collaborative space. The system is highly based on the eXplainable AI Planning paradigm, making the outputs of the planning process more palatable to humans.

Mr.Jones responsibilities in a meeting are divided in two processes, as seen in Figure 2.8:

- Engage, that consists in monitoring various inputs from the work to situate itself in the context of the group interaction. This is achieved by using inputs like speech transcripts, live images, and the positions of people within a meeting space. Using this, the assistant can (1) requisite resources and services that may be required to support the most likely tasks; (2) visualize the decision process; (3) summarize the group decision-making process.

- Orchestrate, providing support in the decision-making process. This is done by using standard planning techniques and falls into four actions, as shown in Figure 2.8 under "Planning". The assistant (1) execute, performing an action or a series of actions related to the task at hand; (2) critique, offering recommendations on the actions currently place; (3) suggest, suggests new decisions and actions that can be discussed; (4) explain, explains its rationale for adding or suggesting a particular decision.

The central component of the system is the Orchestrator, which regulates the flow of information and control flow across the modules, as shown, in the bottom right of Figure 2.8. These modules are (1) processing sensory information from various input devices; (2) handling the different services of CEL; (3) services that attach to the Mr.Jones module.

To communicate the system uses the interface shown in Figure 2.9. It consists of five widgets:

- The largest one on the top shows the different use cases that the CEL is currently set up to support and represents the probability of the distribution that indicates the confidence in the respective task that is currently being collaborated on.

- Top left presents a word cloud representation of its belief in each of the tasks, the higher the size of the word, the higher the probability associated with that task.

- Top right shows the agents that are recognized as being in the environment using four independent cameras.

- The bottom left presents the four camera feeds.

- The bottom right represents a word cloud-based summarization of the audio transcription.

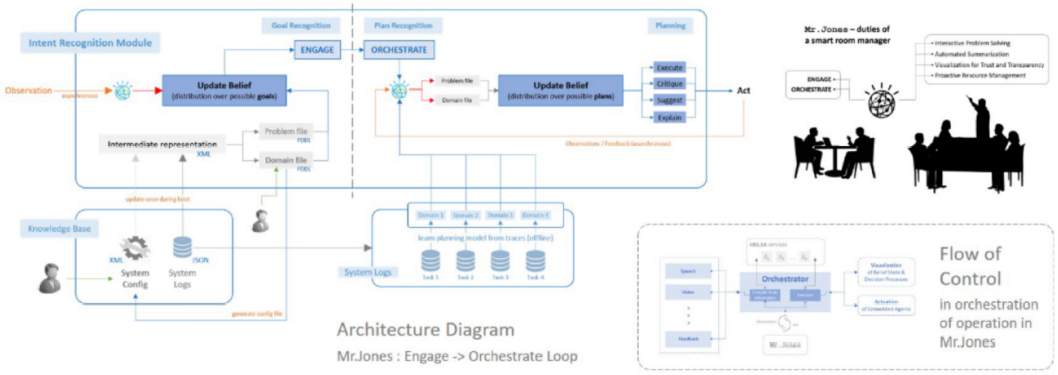


Figure 2.8: Mr.Jones architecture (from [8])

Figure 2.9: Mr.Jones interface (from [8])

2.2.7 Analysis of related work

From the analysis of the related work, some key features were extracted and used to compare the systems in the Tables 2.1 2.2 2.3. This does not have into consideration the system performance for each feature but only if it is present. The last row of each table represents the developed proof-of-concept system that will be presented later, but already gives a brief idea of where it stands compared to similar work.

System	Real-time processing	Post--processing	Dialog acts	ASR
CALO [61, 58, 59, 39]	✓	✓	✓	✓
Meeting assistant intel [2]	✓	✓		✓
The Meeting Project at ICSI [50]		✓	✓	✓
TalkTraces [9]	✓			✓
ProMETheus [47]	✓		✓	✓
Mr.Jones [8]	✓		✓	✓
Developed System	✓	✓		✓

Table 2.1: Related work comparison - Part 1

System	Emotion	NLP	Online	Audio
CALO [61, 58, 59, 39]		✓	✓	✓
Meeting assistant intel [2]	✓	✓		✓
The Meeting Project at ICSI [50]				✓
TalkTraces [9]		✓		✓
ProMETheus [47]	✓	✓	✓	✓
Mr.Jones [8]	✓	✓		✓
Developed System	✓		✓	✓

Table 2.2: Related work comparison - Part 2

System	Other sensors	Engage	Languages
CALO [61, 58, 59, 39]	✓		English
Meeting assistant intel [2]			English
The Meeting Project at ICSI [50]			English
TalkTraces [9]			English
ProMETheus [47]			English
Mr.Jones [8]	✓	✓	English
Developed System		✓	Any

Table 2.3: Related work comparison - Part 3

Comparing the systems they all have a similarity, the audio as a sensor, automated speech recognition, and a limitation of the supported language, as they only support English. The developed system follows the norm, but with the exception of not being language restricted.

One feature that most of the systems focused on was real-time processing at the expense of post-processing. This shows that although post-processing could be helpful to analyze a meeting after its realization, this type of system would have a bigger impact if they can in real-time give information to the user and help in the meeting development.

The use of NLP modules was also highly implemented across the systems, as one key feature on them was meeting summarization. This can also justify the restriction on language since depending on the algorithm used, doing NLP for multiple languages can be difficult and time-consuming.

The less-used features are the online functionality, this can be justified by the time of development of the system, the use of other sensors besides audio, and the lack of engagement. Although all of them in some way give useful information to the user that can help in the meeting development, only Mr.Jones is an active part of it.

The most complete system is without a doubt Mr.Jones, it fulfills most of the features presented in the tables above, it is fairly recent and tries to eliminate what the other systems had as a limitation. The most important feature implemented is engagement, Mr.Jones was the only one capable of being an active part of the meeting, aiming to actively assist in the meeting giving suggestions, critics, and also requisite resources and services that may be required.

Chapter 3

Scenarios and Requirements

> Most meetings are too long, too dull, too unproductive – and too much a part of corporate life to be abandoned.
>
> *(Lois Wyse)*

To drive the current work it was adopted a user-centered Design [23] approach. Starting by identifying the target users and developing Personas and usage scenarios. This enabled a derivation of a set of requirements. The requirements were then used to create a system with the desired capabilities.

3.1 Personas and Goals

For the creation of personas, it was considered a tech company environment, a place where meetings are part of everyday life and where a meeting system would have more impact and be more easily adopted. The main Persona for the work was a project manager, presented as "Pedro". Alongside him are two other personas, "Joana" and "Francisco". They not only will have different needs but will also help to latter better explain the scenarios.

3.1.1 Pedro, Manager

Age: 45
Job: Project Manager
Family: Married and with two childrens
Location: Porto, Portugal
Education: Master's degree in "Management and Planning"

Bio

Pedro Sousa was born on February 23, 1985, and lives in the city of Porto with his wife, Julia Pinheiro, and their children, Filipa and Ricardo, 3 and 8 years respectively. He recently joined a large tech company in Aveiro as his Project Manager, splitting his work between an hour and a half trip to work in person and working from home using online platforms.

Since a very young age, he has always shown pleasure in organizing project meetings and distributing work among his classmates. As a Project Manager, his job consists of that, organize teams and distributing the work among the different members. For that a high number of meetings has to be carried out, this brings a high level of responsibility and difficulty to his job.

Organizing a small team is quite a job already, but when he decided to join the Aveiro company nothing made him expect the magnitude of the projects and the size of the teams he would have to organize. Especially during meetings, it can be quite difficult to keep track of everything.

Goal: Give Pedro tools that will facilitate his coordination job during the meetings, and also give him information that could be helpful to better plan the next one.

3.1.2 Joana, Front-End Developer

Age: 24
Job: Front-End developer
Family: Single
Location: Aveiro, Portugal
Education: Master's degree in "Computer Engineering"

Bio

Joana Filipa was born on January 2, 1997, in Aveiro where she still leaves with her parents. She recently finished her Master's degree in "Computer Engineering" and a few weeks ago got her first job as a Front-End Developer in the same company as Pedro.

In her job, she works closely with other Front-End developers to bring excellent user interfaces to the company's products. To achieve this several meetings had to be carried out where the team discuss different designs and ideas.

Joana is naturally nervous and shy and due to the dimension of the company and her lack of experience, this was accentuated. She is excellent at what she does, but her shyness during the meetings does not let her expose her qualities and ideas for the different projects of the company.

Goal: Give Joana tools that will encourage her to be more proactive during meetings and also notify Pedro of the lack of interaction.

3.1.3 Francisco, Team leader

Age: 36
Job: Team leader
Family: Married
Location: Aveiro, Portugal
Education: Ph.D. in "Computer Science"

Bio

Francisco Fernandes was born em December 5, 1984, in Algarve but moved to Aveiro in the last week with his wife Patricia. Francisco has a Ph.D. in "Computer Science" and already has 15 years of experience.

He is been working all his career as a back-end developer, and although he loves his job he always had a leader mentality. His dream was always to one day have the opportunity to be a Team Leader.

Last week he finally had the chance to fulfill his dream. Pedro's company was looking for a new Team Leader for the Back-End team, and Francisco was chosen. Francisco will work close not only with the back-end team, but also with Pedro, giving information about the development, and also working with the Team Leaders from other departments.

Being a Team Leader is not an easy job, and for Francisco, it is his first attempt at it. He will be joining a project right in the middle of its development, this will bring an even higher difficulty.

Goal: Give Francisco tools that will facilitate his integration into the team and the project that he will lead.

3.2 Scenario(s)

After knowing the target user several scenarios were created. These describe events that affect the personas described above and show how the system helps the user accomplish his goals. These should give a better understanding on what is the main focus of the system and are also the base for the extrapolation of the final requirements.

3.2.1 Scenario 1

Pedro and half of his team arrive at the company for the usual morning meeting. Due to COVID-19, the other half is doing remote work.

Pedro as the Manager greets everyone and starts to ask what was the progress in last week's and each person then proceeds to present their work.

While the participants speak, a meeting management **system is recording all utterances from every participant** [→ $REQ1$] and **processes it** [→ $REQ2$].

During the meeting, Pedro accesses the system dashboard to view **meeting statistics like** [→ $REQ3$]: **number of sentences uttered by each person, number of interventions, average time per intervention** [→ $REQ5$], among others. After doing a quick analysis he realizes that one participant is intervening too much compared to the average. Pedro then proceeds to advert him and asks other team members to intervene.

Joana is in the same meeting, and by the end of it she still didn't participate and Pedro didn't realize it. The system detects this irregularity and proceeds to **alert Pedro** [→ $REQ7$]. Before Pedro formally asks her to intervene, Joana already started talking since the system **also informed Joana** [→ $REQ6$], letting her take the initiative.

3.2.2 Scenario 2

Pedro arrives at the company after lunch and needs to have an urgent meeting with the front-end team. A critical bug was found and it should be fixed as fast as possible.

Before scheduling a new meeting, he starts by accessing his **personal page in the system** [→ $REQ14$] to be sure that he doesn't have already a meeting with the team that evening. After realizing that is not the case he starts scheduling a new one.

To do this he accesses the system and selects that he wants to **schedule a meeting** [→ $REQ12$]. Then proceeds to choose a date and finishes by **selecting the members that need to be invited** [→ $REQ12$].

All the invited members are then **notified** [→ $REQ7$] about the meeting and quickly join. Since the **platform supports all the media available** [→ $REQ13$], Pedro proceeds to use the screen share functionality to reproduce the bug.

With all of the information needed the team easily found the problem and by the end of the day a fix was already available.

3.2.3 Scenario 3

Last week Francisco was promoted to Team Leader of the Back-End department, to better understand the project and what were the next steps he is been asking Pedro to help him.

Pedro accesses the system and **schedules a meeting with Francisco** [→ $REQ12$] for the next day. Francisco then gets **notified** [→ $REQ7$] and the meeting is added to his **personal page** [→ $REQ14$].

On the next day, right before the meeting start, Francisco and Pedro are **notified** [→ *REQ*7] so they don't miss it. Pedro starts the meeting by congratulating Francisco on his new job and asks in what can he help. Francisco responds by asking if there are any documents or information that he could share with him about the project and the team. Pedro proceeds to explain that the company has been using a system that **processes all the meetings** [→ *REQ*2]. It **transcribes all the audio** [→ *REQ*4] and **makes a characterization of the meeting** [→ *REQ*8], showing the topics that were most talked alongside some **individual statistics from all the participants** [→ *REQ*5]. To share this information Pedro **downloads it as a PDF file** [→ *REQ*11] and shares it via email.

With almost no effort Pedro shared all of the information necessary to Francisco, even information that he didn't remember.

3.2.4 Scenario 4

Pedro arrives at the company in the morning and first, he starts his day by **verifying if he has any scheduled meetings** [→ *REQ*14]. He quickly realizes that he has a meeting with the front-end team but he can not quite remember what was done in the past week.

Pedro accesses the system and proceeds to analyze the meeting dashboard. Since he does not have the time to visualize all of the information he instead uses the **chat-bot functionality** [→ *REQ*10]. He asks for a resume of each participant's work and what was planned to be done that week. The bot quickly responds and Pedro now has a clear idea of what he needs to ask in the meeting later that morning.

3.2.5 Scenario 5

Emotions can have a big impact on how we work, and to achieve high productivity everyone should be in a good emotional state. Pedro as a Manager wants his teams to be as much productive as possible and assist everyone in their needs.

During a meeting, Pedro **observers the dashboard** [→ *REQ*3] and realizes that **Joana is demoralized** [→ *REQ*9], and that **she is not participating as usual** [→ *REQ*5]. This emotional information is **private and only available to the meeting coordinator** [→ *REQ*6, 14].

Instead of bringing this information up during the meeting, he waited until the meeting was over. In private Pedro had a quick chat with Joana, understanding what she was going through and if he could help in anything.

3.2.6 Scenario 6

Before remote work got implemented, Pedro was used to having presential meetings. The most important ones already had in place a recording system, where it was recorded the audio from each participant individually. This was useful but time-consuming if he wants to search for something in particular.

Using the systems **pre-recorded meetings feature that lets users upload full meetings** [→ *REQ*15], Pedro takes the files and uploads them. After some minutes **all the information produced is ready to be analyzed** [→ *REQ*3] as if it had happened inside of the system.

3.3 Requirements

The table 3.1 presents the requirements extrapolated from the scenarios described above. The last column is the level of importance being 0-Extremely Important, 1-Very Important, and 2-Moderately Important. This was done taking into consideration its overall importance and impact on the final system.

Req nº	Requirement	Importance
1	Recording and storage of all interventions from the audio of each participant, clearly and well identified.	0
2	Process all the audio gathered, no audio should be lost or ignored by the system	0
3	Easy access to the information gathered and produced by the system,	0
4	Speech-to-text, transcribing all the audio gathered	0
5	Meeting statistics, produced from analysing the audio or the transcription	0
6	All the meetings participants have access to the platform but with different roles associated	2
7	Notification system that pushes notifications to specific users and the meeting coordinator, making it a active part of the meeting	1
8	Make a characterization of the meeting, showing the most addressed topics and what was defined by the end of the meeting	1
9	Give the associated emotion from a meeting participant using the audio available	1
10	Chat-bot, a entity where the participants can ask questions and the bot will respond based on the information gathered	2
11	PDF generation of the information produced, used to better share the information between users outside of the system	2
12	Meeting scheduling and participant invitation	2
13	Meeting realization inside the platform supporting audio, video and text chat	2
14	User identification, each user has is unique login, unique meetings that can access or get invited to and even different roles inside the meeting.	2
15	Pre-Recorded meeting analysis with audio serialization	2

Table 3.1: System Requirements

Chapter 4

Tools and Technologies

> You have a meeting to make a decision, not to decide on the question.
>
> *(Bill Gates)*

In this chapter, all the tools and technologies used throughout the development of the system are presented. In Figure 4.1 it is possible to see the list of all the chosen ones. Through the chapter will also be explained the process behind their selection, comparing in some cases to similar technologies.

Figure 4.1: Tools and technologies used throughout the development of the system

4.1 Meeting Platform

Working from home and online meetings have become a standard and widely adopted. According to a global survey conducted by Gartner, Inc [48], 88% of the organizations, worldwide, made it mandatory or encouraged their employees to work from home after COVID-19 was declared a pandemic.

It is clear that this way of working is here to stay so any system developed to help and improve the participant experience in meetings should be able to work in this environment.

Several online platforms provide the capabilities to hold an online meeting. The first platform tried was Zoom [67], a proprietary video teleconferencing software program developed by Zoom Video Communications. According to Zoom's statistics [27], the platform has now over 300 million daily meeting participants, an increase of 2900% since December 31, 2019.

Zoom provides tools to stream meetings [68], but with some limitations. The audio from each participant does not have its own track in the stream or the recording, making it difficult to identify users. Being a proprietary solution means that any system built around it is constrained to what the company has available.

The search for systems not suffering from those limitations resulted in the identification of Jitsi. Jitsi [41] is a free and open-source web-based application, for voice, video conferencing, and instant messaging. It is multi-platform and well-documented [41, 18]. It received support from some well-known institutions such as the University of Strasbourg and the European Commission. Being an open-source platform means that it is possible to host a version of it in our servers and change it to the user's needs. They also have a version of it in their own servers at `https://meet.jit.si/` and since their solution already has the features needed it was chosen.

Jitsi also provides a JavaScript library, "lib-jitsi-meet" [16], that enables interaction with the JITSI platform. It allows the creation of a participant inside the meeting that has access to everything, including the audio from each participant individually. Since it is an open-source platform it was possible to change this library to be able to record the audio and send it to a server to be stored.

The overall architecture of Jitsi is presented in Fig.4.2. The main components are [17]:

- **Jitsi Meet**, a JavaScript application that uses WebRTC and the Jitsi Videobridge to provide high-quality video conferences. It was built using React and React Native.
- **Jitsi Videobridge**, designed to route video streams amongst participants.
- **Jitsi Conference Focus**, component used to manage media sessions and acts as a load balancer between the participants and the videobridge.
- **Jitsi Gateway to SIP**, allows regular SIP clients to join Jitsi Meet conferences.
- **Jitsi Broadcasting Infrastructure**, set of tools for recording and/or streaming a Jitsi Meet conference.

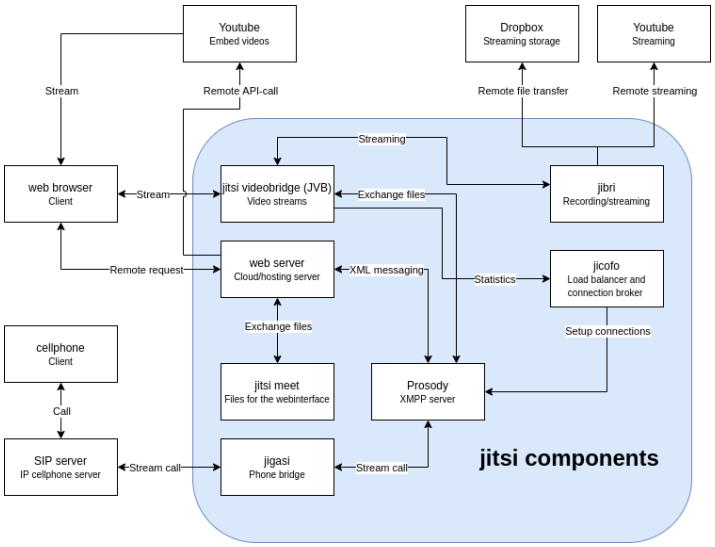


Figure 4.2: Jitsi Architecture (from [17])

The architecture of the system, which will be presented further in this dissertation, was built in a way that any solution to capture audio from a meeting would work even in an in-person scenario. Since the actual conjecture would make testing difficult a fully online approach was implemented.

WebRTC is the tool used by Jitsi [41, 18] to allow voice and video conferencing. It is a free and open-source project that provides tools to add real-time communication capabilities to an application. It supports video, voice, and generic data to be sent between peers and is available on all modern browsers as well as on native clients for all major platforms. It is supported by Apple, Google, Microsoft, and Mozilla, amongst others [22].

Figure 4.3 shows how the connection between two users is done. To create the connection between Alice and Bob a web server is needed to signal the path. After both users know the path of each other, the media path is a peer-to-peer connection. Since the server has no access to the media being produced by the participants, the "lib-jitsi-meet" library has to create a participant.

To record the audio the system uses RecordRTC [20]. A library used to record audio, video, screen, and canvas from a WebRTC connection.

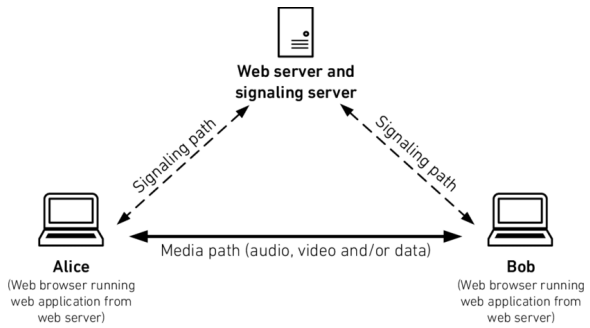


Figure 4.3: WebRTC [22]

4.2 Speech and Language Processing

One important requirement for the system was the capability to do Speech and Language Processing on the recorded audio. To achieve this two technologies were used. The first one is focused on transcribing the meeting into text. This is done using an Automatic Speech Recognition (ASR) service. The second one is focused on producing analytical information regarding speech using the Python library, "My-Voice-Analysis".

4.2.1 Speech-to-Text

The capability to transcribe the audio gathered from the meeting, was an important requirement for the system since the beginning

For this, an Automatic Speech Recognition (ASR) [29] service is needed. ASR enables the recognition of spoken language to text by computers, which can be speaker dependent or independent, depending on if they require to be trained for each user. Since the platform is to be used by multiple participants the service has to be speaker independent.

To achieve this two services were tested, IBM Watson Speech to Text [38] and Google Speech-to-Text [34]. Both present similar features and use machine learning for automatic speech recognition. But there are two differences.

The first one is the number of supported languages, it is limited in the IBM solution compared to the Google. According to their specific documentation, the Watson Speech to text supports a total of 19 language models [36], in contrast to the Google service that supports over 125 languages and variants [34].

The second difference is performance and accuracy. The IBM Watson Speech to Text only supports the processing of one file at a time and in tests showed to be less accurate and confident in his results compared to the Google solution.

It should be noted that both services support customized speech recognition to transcribe domain-specific terms and rare words to boost transcription accuracy.

4.2.2 Voice analysis

Having access to all the audio individually from each participant means that it can be processed and it is possible to have a better understanding of how each participant speaks. It is important that this processing is done directly on the speech audio. This way it is possible to obtain information regarding speech mood, articulation rate, speech rate, speech time, between others, without being language constrained.

A solution was built using a Python library, My-Voice-Analysis [45], developed by Sab-AI Lab [46] in Japan for the analysis of voice without the need of a transcription. The processing of the audio breaks utterances and detects syllable boundaries, fundamental frequency contours, and formants [45]. It is a unique tool as it aims to provide a complete quantitative and analytical way to study acoustic features of a speech.

The library was developed based on studies carried out by Nivja DeJong and Ton Wempe [26], Paul Boersma and David Weenink [4], Carlo Gussenhoven [35], S.M Witt and S.J. Young [66] and Yannick Jadoul [40].

4.3 Database / Knowledge base

The targeted system is highly based on producing and processing information, this data needs to be stored in a way that can easily be accessed by all the services.

In the early stages of development, a relational Database [55] was used. Each component of the system had its own table, each row was a unique record identified by a unique ID and each column was an attribute of that record. These tables could be related to each other using the unique ID that identifies them. For this it was used MySQL, which is the most popular open-source database, used by the world's largest organizations including Facebook, Google, and Adobe, it is well documented and with excellent performance.

At the time it looked like the best option but when the system grew it showed its limitations. The way that the system was developed it started to be noticeable that it was hard to keep track of everything and connect all the different tables to each other. For a small meeting, a big number of queries were already been used to achieve what it seems like could be done in a couple.

Instead of the relational Database, it was taken a different approach and the idea of treating the information if it was an object [56] began to take place.

The Database should be able to link all the data stored, knowing the relations between all the objects presented. This concept is called by the W3C as Semantic Web [63]. The knowledge base was then implemented and uses technologies such as RDF, SPARQL, and OWL.

Resource Description Framework (RDF) [62, 52] is the standard model to interchange data in the Web. Developed and standardized with the World Wide Web Consortium (W3C), allowing multiple schemas to be applied, interlinked, queried as one, and modified without changing the data instances. It works by using triples (Figure 4.4), this means that the Database has only one table with three columns. The first column represents the subject, for example in a user context the subject will be the user identification, the unique ID from the Relational Database [55]. The second column is the predicate, in the user context would be for example "name". The last column is the object, in the user context will be the actual name of the user, for example "João". All data can be converted to RDF with no exception.

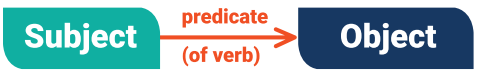


Figure 4.4: Triple [52]

SPARQL [62, 53] is the standard query language and protocol that enables the user to query information from any database or data mapped in RDF. Developed and endorsed by the W3C it allows users to retrieve, store and modify data mapped in RDF, but can also be executed on any database that can be viewed as RDF via middleware.

Web Ontology Language (OWL) [65, 6] is the ontology language of the Semantic Web. An ontology is created to specify the terms in a domain and the relations between them [51]. The main goal is to share a common understanding of the structure of information among people and computers.

Protégé [60, 25] is a free open-source editor and framework based in Java to create ontologies, developed by the Stanford University. This tool was used to develop the system's ontology.

Building an ontology even with this type of platform is not easy as they are designed for those in the field of ontology and with some degree of knowledge about the underlying axioms. This platform was chosen, having in mind the wide adaptation and active community behind it, meaning that even with little knowledge on the subject it was easy to find examples and documentation explaining all the process.

Also the document "Ontology Development 101: A guide to creating your first ontology" [51] it is a recommended reading since it was written by researchers at the Stanford University and was the perfect starting point.

Virtuoso Universal Server [54, 64] developed by OpenLink is a hybrid Web Application Server that provides SQL, XML, and RDF data management in a single multi-threaded server process, available in an open-source and commercial version. It is a powerful platform with an extended list of features. The system shifted to the Semantic Web and Virtuoso covered all the needs. It does RDF data management, has an OWL Reasoner, meaning that the created ontology could be easily imported and used, and has a SPARQL query service endpoint that any programming language supports.

4.4 Development of client-server solutions

This section is divided into two parts, server-side and client-side. The first one is focused on presenting how the technologies previously introduced were implemented and how this information is then sent to the final user. The last section is focused on how the information produced by the server-side is presented to the user.

4.4.1 Server-side

Every system needs a solid back-end and there are a lot of technologies and programming languages that can be used to achieve this, most of the time those are constrained by the tools used. It should be noted that because of the adopted decoupled architecture, the back-end is not limited to only one technology or one programming language. The system was built using two main languages, Python and JavaScript.

Python [33] is only used for one module. As explained before the library My-Voice-Analysis [45] is a Python library, making this module necessarily the only supported language.

For the rest of the system, there were no limitations since all of the remaining technologies used had APIs and libraries for all of the most known languages.

Javascript was chosen with the powerful runtime environment Node.js [42], an open-source, cross-platform back-end that runs on the V8 engine. The reason behind this choice was the "JavaScript everywhere" paradigm [24], unifying the development of web-based applications around a single programming language. This means that both server-side and client-side share the same language making development and maintenance easier.

With Node.js [42] it is possible to use the Express [32] web framework, A minimal and flexible platform that makes effortlessly the creation of robust web REST APIs.

4.4.2 Client-side

Interface and usability is an important aspect of any system. One important factor for any system is the way it is presented to the final user, this can be implemented in a variety of ways, from desktop, web, or mobile application, all solutions have their pros and cons.

Since the system uses the Jitsi Meet API library and being a JavaScript library that needs a browser to work the only solution available was a web application. Compared to a desktop or a mobile application, a web application is more versatile, it does not need installation and it is not restricted to the operating system or a device.

For its development, a number of frameworks were considered, all capable of producing high-quality web applications. Having in mind again the "JavaScript everywhere" paradigm [24] the focus was shifted towards frameworks that have JavaScript as their programming language.

In the end, React [31] was chosen, it is an open-source JavaScript library used to build user interfaces or UI components. It is maintained by Facebook, a group of individual developers and companies. Being stable and well documented, it was fast to implement and learn. It is widely used, which means that any problem encountered during the development has a solution online.

Chart.js powers all charts available in the platform dashboard, giving an easier way to present data to the final user.

It is a free and open-source Javascript library used to create interactive charts for web applications.

It supports a variety of charts: bar, line, area, pie, bubble, radar, polar, and scatter, those are rendered inside an HTML5 canvas.

Although this library lacks customization it is significantly easier to use and implement compare to others like D3.js [14]. It also has a well-documented react wrapper, react-charjs-2 [13], that makes the integration into the web application faster and easier.

4.5 Communication between modules

In a system that adopts a decoupled architecture some modules will be producing information, others will be accessing that information and some will do both. This is a use case scenario for the use of a message broker.

A broker [37] works as a middleware enabling real-time communication between different applications, services, and systems. There are a lot of solutions and all of them would work because they all share the main feature publish/subscribe messaging [3], this means that any module can publish information into the broker or consume that information.

For this system Apache Kafka [1] was chosen, it is an open-source solution maintained by Apache and consists of a distributed system and uses a high-performance TCP network protocol.

For the developer level, it is easy to adopt and use. But the key feature behind its selection was that it saves the messages for a period of time. This way if the consumer is not connected to the broker when he connects it will send all the messages that he missed. This was important in the early development stages when the system was unstable.

Figure 4.5 presents an illustration of how Kafka works. Starting with the "Kafka cluster", it is composed of different "Topics", these represent different data streams. Taking a

newspaper as an example, one "Topic" could be directed to sports, the other to finances, and the last one to politics. Each "Topic" can be even further divided into different partitions to improve performance. At the top is possible to see the "Producers". These will be publishing information on a specific "Topic". On the other end are the "Consumers", they are connected to a specific "Topic" and will receive the information when it is published.

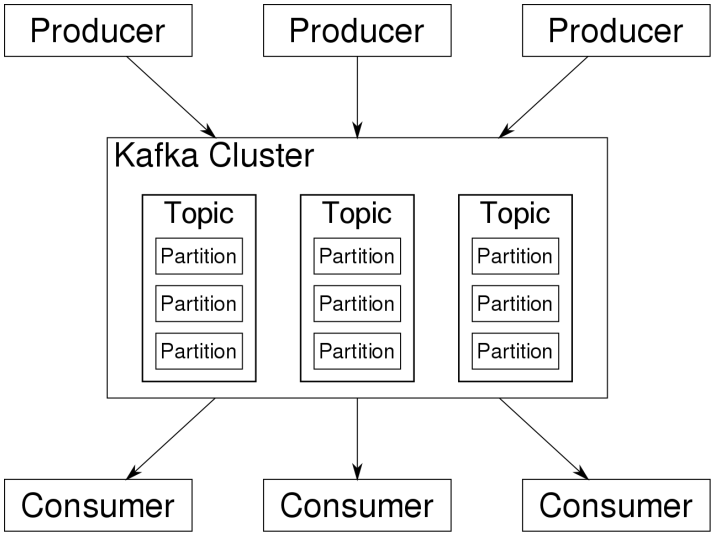


Figure 4.5: Kafka overview

4.5.1 WebSocket [19, 21]

For the communication between the client-side and the server-side, as explained in the section 4.4.1, it was used "Express" to create REST APIs. These have a limitation, the server-side will only return information when the client-side asks for it.

For the Alert module, this would not work, the server-side should be able to send the alert directly to the user.

Kafka was not an option in this case, since it only works between the different modules on the server-side.

WebSocket [19] was chosen, as it enables real-time bidirectional communication between a server and a browser over a single TCP connection. With WebSocket, it is possible to have event-driven responses to messages without the need to poll the server.

To implement WebSocket into the system it was used Socket.IO [21]. Since it is a JavaScript library meant that both sides of the communication share nearly identical APIs, making the development and maintenance of the service easier.

Figure 4.6 demonstrated the connection between a client and a server using WebSocket. To establish a connection, the client sends a Handshake request to the server, which returns a handshake response. Once the connection is established, communication switches to a bidirectional binary protocol. At this point, both client and server can send data back and forth in full-duplex mode. To close the connection it is only required that one of the sides closes the channel.

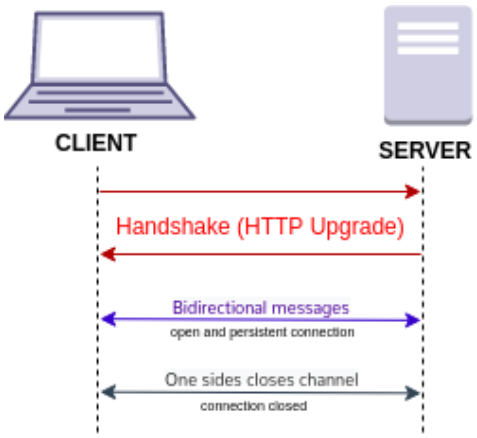


Figure 4.6: Example of a connection using WebSocket

Chapter 5

Proof-of-concept System

> Most people simply don't view going to meetings as doing work.
>
> *(William Daniels)*

This chapter describes the overall architecture of the proposed proof-of-concept system. It starts by giving a brief introduction to the system, providing a conceptual architecture, the main components, and their functionality. In the second section is done a more in-depth look into the implementation of the system, with the detailed architecture and a deep explanation of each module that composes the system.

5.1 General Architecture

This section presents a brief introduction to the system and the most important modules. Figure 5.1 illustrates the conceptual architecture, presenting the modules and their interactions. These modules will be further analyzed in this chapter.

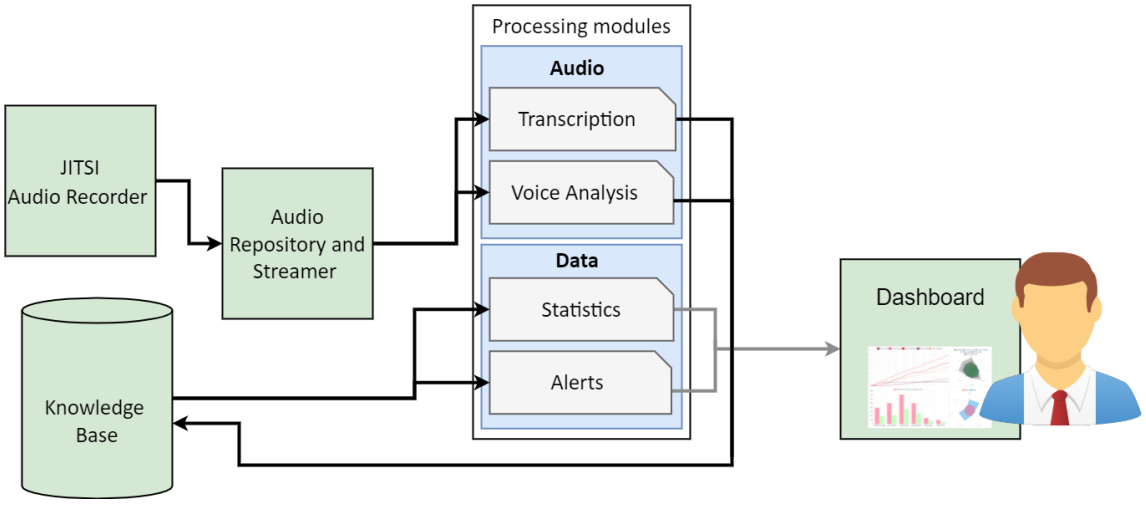


Figure 5.1: Conceptual Architecture

Briefly, the main components of the system are:

JITSI Audio Recorder module that records all the audio from the meeting.

Audio Repository and Streamer module that stores the audio recordings and makes them available to other modules through streaming.

Processors a group of modules that process information. They are further divided into two groups: (1) Audio, representing all the modules that exclusively process audio, and (2) Data, representing the modules that process the data produced by the Audio group.

Knowledge Database a flexible solution for information management using a semantic approach.

Dashboard a web interface where the final user can interact with the system.

5.2 Implementation

To provide flexibility to add or change the modules, the system adopted a decoupled architecture.

Most communications of the system are made using a distributed event streaming platform, in the system some modules are producers of information while others consume it.

This section presents information on how the architecture was instantiated to create a first proof-of-concept and an explanation of each module and its functionality.

Fig. 5.2 presents a global view of the system, with main interactions among modules.

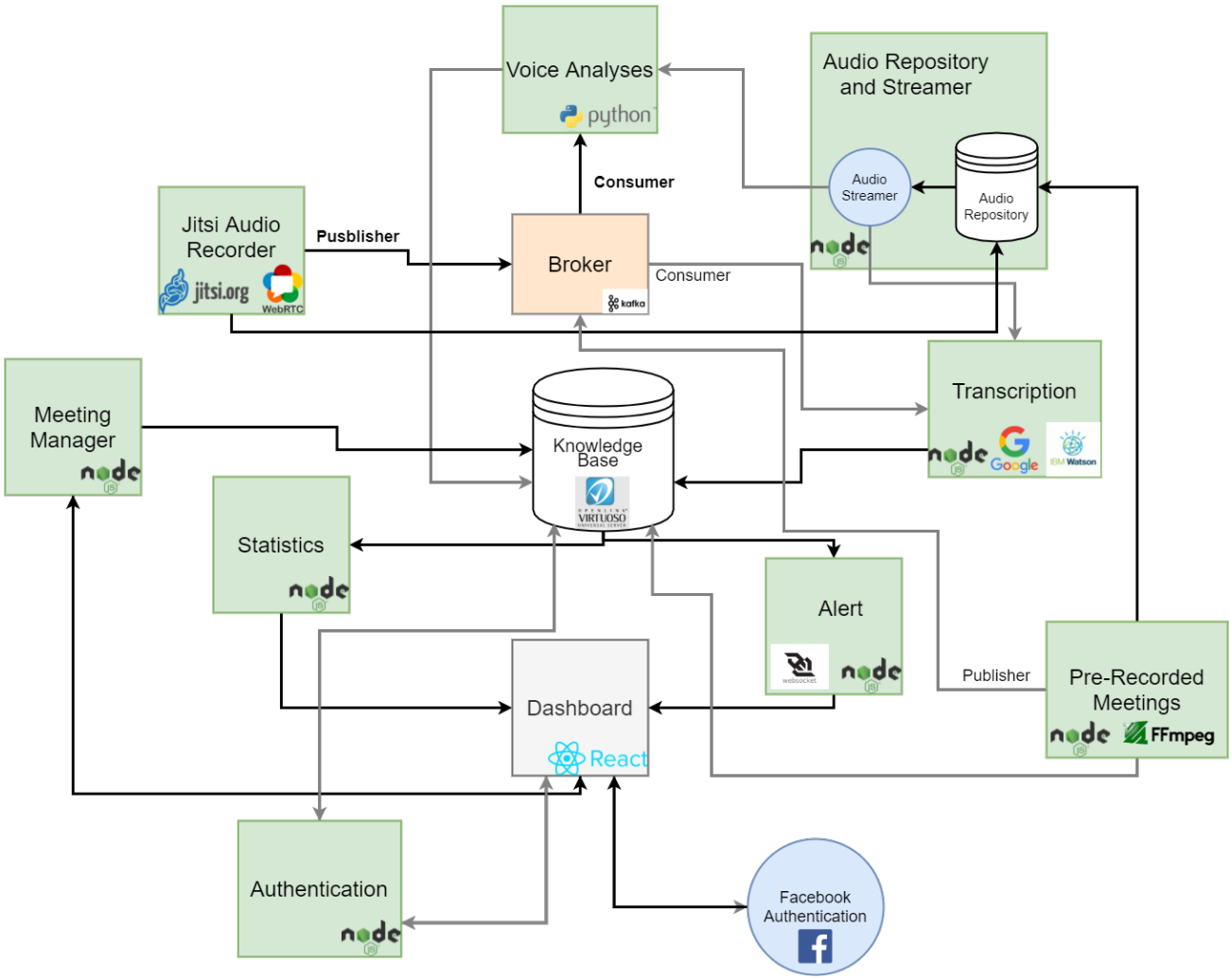


Figure 5.2: Detailed System Architecture

5.2.1 Jitsi Audio Recorder

This module is responsible to record all the audio from the meeting.

To achieve this it is used Jitsi [41] meeting platform and the library lib-jitsi-meet [16].

Some modifications were made to the library to support audio recording. A Voice Activation Detection algorithm was added to mark the beginning and end of the speech. Then since Jisti uses WebRTC, it is possible to use RecordRTC [20] to record the audio segments from each participant, individually.

For each segment, information regarding user and meeting identification is added alongside a timestamp.

The segments are then sent to the Audio Repository that stores them and makes them available to other modules through streaming.

5.2.2 Audio Repository and Streamer

This module is responsible for storing all the audio files produced by Jitsi Audio Recorder and making them available through streaming to other modules.

To achieve this a unique ID is generated for all the audio segments received. A new entry is created in the knowledge base alongside the information already available (participant and meeting identification, and timestamp).

The audio is stored in the Audio Repository using the unique ID as the file name. This information is then published on a Kafka topic.

Other modules can access the audio segments stored using the "Streamer". It requests the introduction of the name of the file and starts streaming it if it is available.

5.2.2.1 Voice analysis

This module is responsible for analyzing the speech from the meeting.

To achieve this, this module works as a consumer on the Kafka topic. Every time new audio is available, it requests the Audio Repository and Streamer module for its stream and stores a copy locally.

Then, it uses the my-voice-analyses library to analyze the audio. This library can produce accurate information without the need of transcription. Its built-in function recognizes and measures:

- gender recognition,
- speech mood,
- pronunciation posterior score
- articulation-rate,
- speech rate,
- filler words,
- f0 statistics,

The information produce is then stored in the knowledge base where it can be further analyzed by other modules.

After all the processing has been done the local copy is deleted.

5.2.2.2 Transcription

This module is responsible for transcribing all the speech recorded by the Jitsi Audio Recorder module.

To achieve this, similar to the Voice analysis module, it works as a consumer on the Kafka topic. Every time new audio is available, it requests the Audio Repository and Streamer module for its stream and stores a copy locally.

This file is then sent to a service that transcribes the audio into text. For this matter two services were tested: IBM Watson Speech-to-text [38] and Google Speech-to-text [34]. Both return the transcription alongside its confidence.

The Google service has shown to have better results for European Portuguese and also supports a higher number of languages.

All the information generated is stored in the knowledge base and can be directly presented to the user and further analyzed by other modules.

After all the processing has been done the local copy is deleted.

5.2.2.3 Statistics

This module provides statistics regarding analyzed meetings. These can be directed to users, individual meetings, or even carried out globally.

To achieve this the module takes advantage of the information produced by the "Voice analyses" and "Transcription" modules, and the knowledge base, using SPARQL queries to produce the information needed. When the complexity of the statistic goes beyond what is possible to do only with a query, it does some processing to achieve the final result.

This information is then sent to the dashboard where it is displayed in plots or simple text to be analyzed by the user.

With this module, it is possible to compare values between different participants such as intervention time, analyze the development of the meeting to verify when it promotes greater intervention and discussion between the participants, and others. It is also possible to have more global statistics such as average intervention time by participant.

This can give useful information to the meeting coordinator, helping him achieve a better meeting.

5.2.2.4 Alerts

This module provides the system with alerts and notifications. These are divided into two types: (1) Performed by the meeting coordinator, and (2) Automated Alerts.

The first type of alerts are correlated to activities performed by the meeting coordinator. These are available inside the main meeting page through the quick access menu (Figure 5.3).

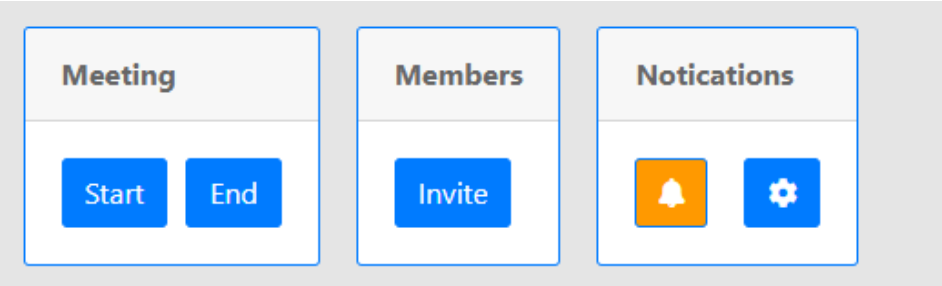


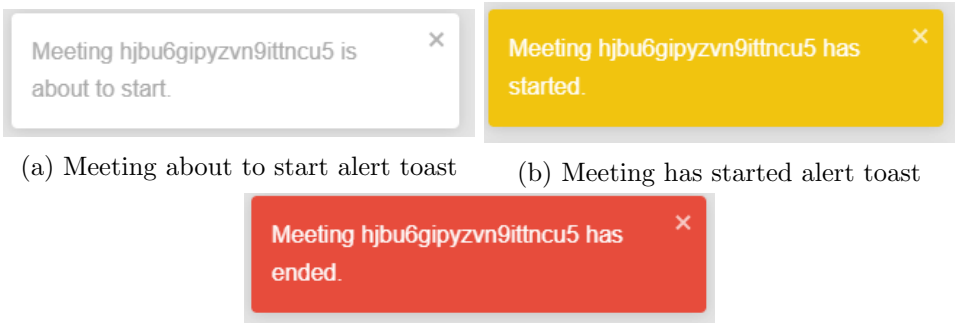
Figure 5.3: Quick access menu for the meeting coordinator

Pressing the orange button with a bell will notify all the meeting participants that the meeting will start soon. This is useful if the meeting coordinator realizes that several participants are still missing. Pressing the alert toast (Figure 5.4a) with the left button will redirect the user to the specific meeting page.

Pressing the start button does two things. First, it sends a notification to all the participants informing them that the meeting has started (Figure 5.4b). Lastly, it activates the

automated alert system. The alerts that this module can produce will be presented further in this dissertation. Pressing the alert toast with the left button will redirect the user to the specific meeting page.

Pressing the end button does also two things. It sends a notification to all the participants informing them that the meeting has ended (Figure 5.4c) and also disables the automated alert system.



(a) Meeting about to start alert toast

(b) Meeting has started alert toast

(c) Meeting has ended alert toast

Figure 5.4: Group of alert toasts performed by the meeting coordinator

The second type of alerts works by periodically querying the knowledge base for information that will be useful for the development of the meeting. A detailed explanation of how this is done will be presented further in the dissertation. Some examples of these alerts are low transcriptions confidence, alert participants that are talking too much or not enough. These can be activated/ deactivated and configured using a user interface (Figure 5.5). It is accessible by pressing the settings button under "Notifications" on the quick menu.

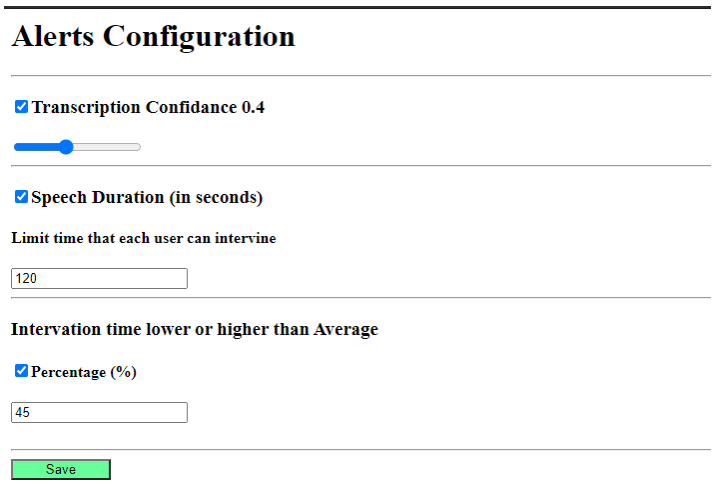


Figure 5.5: Alert Configuration Interface

The module also provides a REST API, enabling other modules to send notifications to a unique user or all.

It uses WebSocket [19] to directly push an alert to the end-users in combination with a cache that will store the notifications until the user sees and removes them from the system.

5.2.3 Knowledge base

To create a flexible solution for information management that would support complex queries and also give an easy way to introduce new modules into the system without an overall design of the database a semantic approach was adopted.

The development consisted of 3 parts, to be detailed in this section: (1) a small domain ontology; (2) selection and use of a triple store; and (3) queries in SPARQL.

5.2.3.1 Ontology

The developed domain ontology was built, as explained before, using "Protégé" [60, 25] and integrates 5 classes: Meet, User, Audio, Transcription, and Statistics.

Relations among classes and classes' properties are summarized in Fig. 5.6

The ontology was easily imported into the knowledge base using the 'Virtuoso Universal Server" web interface.

In the future, it is certain that this ontology will need to be revised and most likely this process will continue through the entire lifecycle of the ontology [51] as its development is necessarily an interactive one.

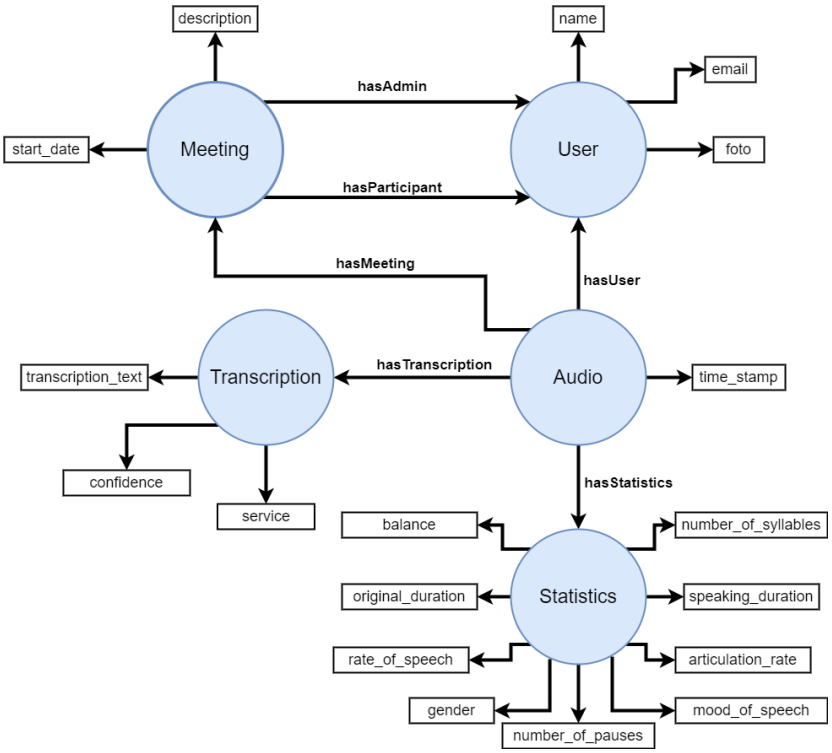


Figure 5.6: Domain Ontology developed, showing classes their relations and properties

5.2.3.2 Triple Store

The knowledge base uses the World Wide Web Consortium (W3C) standards, using Resource Description Framework (RDF). This was better explained in section 4.3.

To store this type of data the system uses "Virtuoso Universal Server" a hybrid Web Application Server that provides SQL, XML, and RDF data management in a single multi-threaded server process.

5.2.3.3 SPARQL Queries

To store and gather information in the knowledge base the system uses "SPARQL", the standard query language and protocol for databases or data mapped in RDF.

Each system module stores or retrieves information using the "Virtuoso Universal Server" SPARQL query service endpoint. This is supported by any programming language removing any restriction that could appear in this regard.

5.2.4 Auxiliary

This section presents auxiliary modules. This modules do not have a high impact on the system as they do not participate in the stream of data as producers nor as consumers. They were created to give a better experience to the user or to help better organize data.

5.2.4.1 Authentication

The module is responsible for authenticating and registering new users into the system.

This is done using the Facebook Authentication system [30] to validate the user. If valid and it is the first time the user is accessing the platform a pop-up is shown, requesting access to some personal information. This information is sent to the authentication module where a unique ID is generated and stored in the knowledge base alongside the information gathered from Facebook.

If the user is already registered in the system, after the Facebook Authentication validation, the personal information is sent to the authentication module to find the match ID. This ID is sent to the user and used to access all the system functionalities.

It should be noted that Facebook Authentication is being used as an example, and proprietary or other third-party authenticators should be fairly easy to implement.

5.2.4.2 Meeting Manager

The module is responsible of creating meetings and their management.

To create a new meeting the user has to fill in the information of date and time of the start of the meeting and a description, as seen in Figure 5.7. A unique ID is then generated for each new meeting, and all of this information is stored in the database.

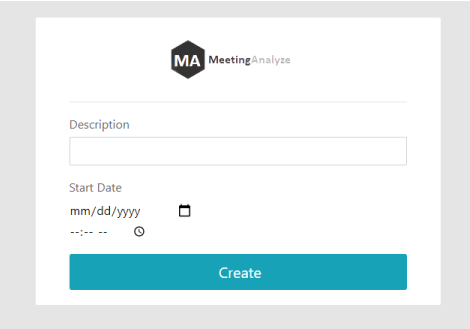


Figure 5.7: New meeting creation

His creator is then assigned as the meeting admin and can invite other participants using their user ID through the meeting page.

5.2.4.3 Processing of Pre-recorded meetings

Although the platform was built for real-time meeting processing, enabling the processing of previously recorded meetings was a must.

This module creates an environment where a full meeting is split into segments and treated as if it was in real-time.

To achieve this it uses FFMPEG [15], a framework capable of decode, encode, transcode, mux, demux, stream, filter and play almost every type of audio file available. With this framework is possible to define a threshold of noise where it is considered by the system as an intervention and a minimum silence length, so interventions would not be cut while the user is speaking.

The segmented audio files are stored and for each, an id is assigned. An entry is added to the knowledge base with the user, time/date, and meeting information.

Similar to the Audio Recorder module, these files are then published into a Kafka topic and will be treated in the same way as the ones obtained in real-time.

This not only gives an easy and fast way to introduce a lot of data into the system but also shows its versatility. The module was added later in development, and it proved the importance of the decoupled nature of the used architecture.

5.2.5 DashBoard

This module provides an interface where the final user can interact with the system.

The dashboard is a separate web application, using React [31], that accesses all the functionalities of the system. This removes any restriction that could appear of the operating system or device type. It can run in all major browsers, this means that it can run even on mobile although it is not fully optimized for it.

To achieve a better-looking application a template [10, 11] was used although highly modified to better suit the needs of the system. The free open-source JavaScript library Chart.JS [12] was also used for data visualization.

The communication to the server-side is done using a “Rest API” or “Webscokets”.

Main windows implemented are:

- Active meetings (Figure 5.8);

- Meetings history (Figure 5.9);
- Meeting Statistics (Figure 5.10);
- Meeting page (Figures 5.11a and 5.11b).

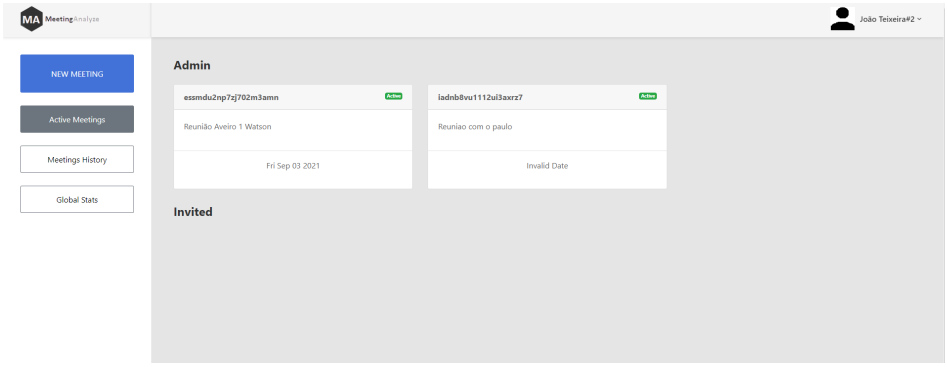


Figure 5.8: Active meeting tab

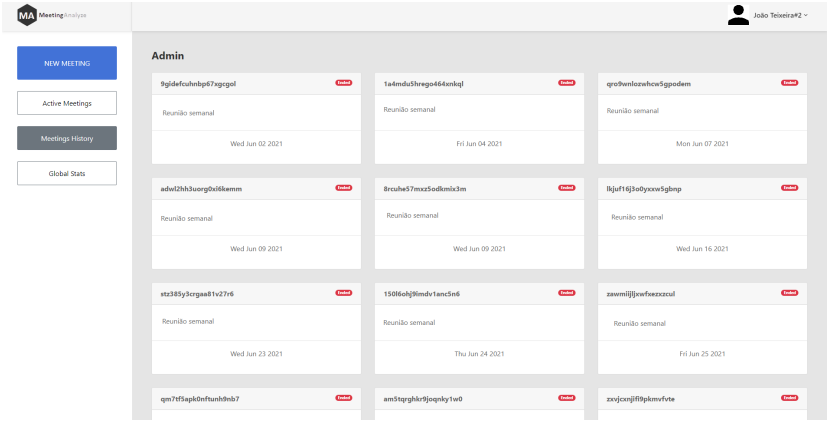


Figure 5.9: Meetings history

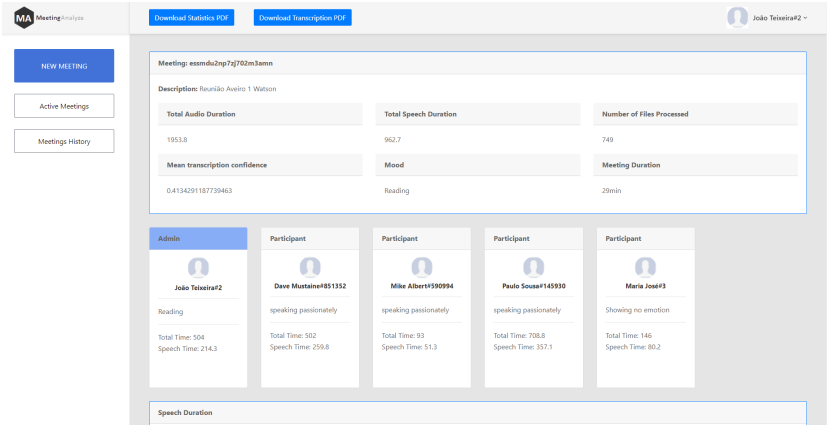
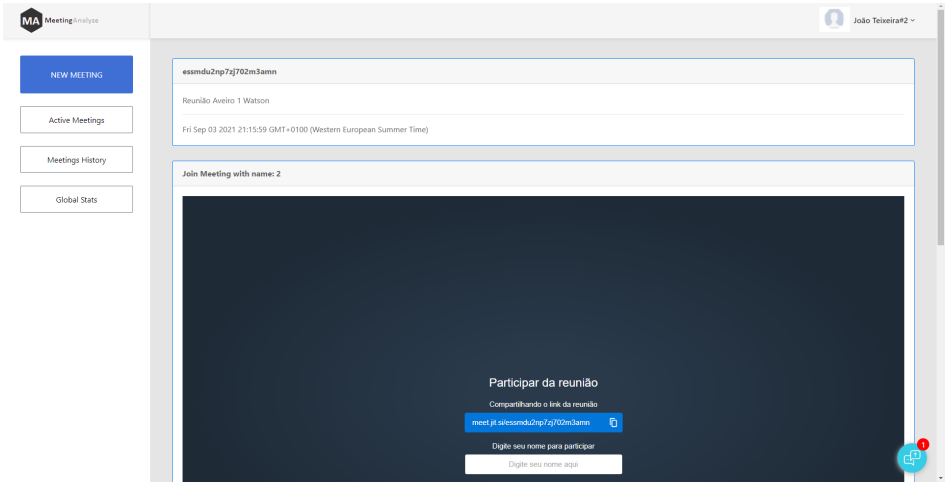
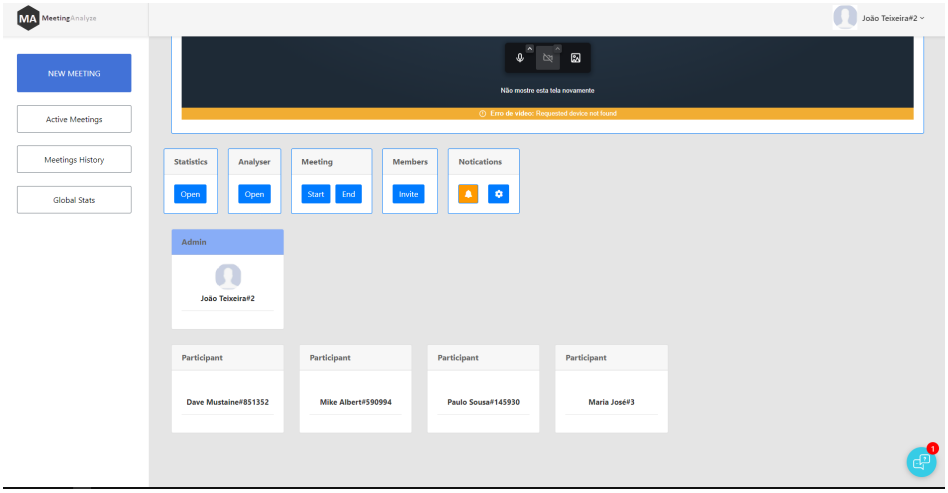


Figure 5.10: Statistics page



(a) Meeting page #1



(b) Meeting page #2

Figure 5.11: Meeting page

Chapter 6

Results

> The meetings can be a lot of fun or they can be frustrating.
>
> *(Bob Weir)*

In this section, the capabilities of the developed proof-of-concept are illustrated. To achieve this a set of pre-recorded meetings was used.

These meetings refer to a social experiment carried out at the Universidade de Aveiro in which the participants organize themselves in groups of four to perform a collaborative task. The data set used is composed of 120 hours of audio, recorded from each participant individually using a personal headset and by the coordinator of the experiment. For this work, five meetings were used combining around 10 hours of recorded audio.

Table 6.1 shows the number of meetings, their duration, number of participants, and number of generated triples.

Nº	Duration	Participants	Number of generated triples
1	14 min	4	4800
2	20 min	5	5325
3	23 min	5	19175
4	36 min	5	24474
5	29 min	5	15729

Table 6.1: Meetings processed alongside duration, participants, and number of generated triples

There is a noticeable difference in the number of triples generated between the first two meetings and the other three. This is justified by a change in the audio segmentation algorithm. In the first two meetings, the quality of the data produced was considered poor. Although the number of generated triples increased exponentially, it had no implications for system performance and produced more reliable and quality data.

The presented system can extract a set of statistical parameters, which will be presented in the form of charts or text. These were divided into three main categories.

The first one represents results gathered from the analysis of a single meeting.

The second category shows the possibilities of analyzing data from multiple meetings.

This is further divided into two other categories, one related to meetings and the other to users.

The last one shows the potential of the system in taking part in the meeting. It uses the implemented alert module to give in real-time useful information to the participants.

6.1 Meeting statistics

This section illustrates the system's capabilities in producing statistical information from a single meeting. From the data set previously introduced was chosen the meeting number 5 as it had shown to have the most interesting results.

Profiting from using a knowledge base, most presented data is directly extracted from some complex SPARQL queries. When it is the case, to better understand how it is done some queries will be presented alongside an explanation behind their development.

In some rare cases, the information needed goes beyond what it is capable of being done using only a query. In these cases, some local processing has to be done on top of the information retrieved.

In the current implementation, the system is capable of producing 4 sets of data.

The first set shows the evolution of speaking time for each participant throughout the length of the meeting.

The second one presents the total speaking time and innervation time for each participant as also a meeting average.

The third set presents a multidimensional analysis of the meeting, comparing several parameters from each participant to a meeting average.

Lastly, the last set shows the mood of speech for each participant individually.

This information is mainly presented in plot form a plot on the system meeting dashboard.

Although the results presented in this section are from the complete meeting, the system does the processing in real-time. The dashboard is accessible at any time to see the current progress, as information is displayed as soon as it is available.

6.1.1 Individual speaking time through the duration of the meeting

The plot presented in Figure 6.1, produced by the system shows the evolution of the total speaking time of each participant throughout the meeting. This visualization provides an easy way to understand the momentum of the meeting. It allows users to analyze when the meeting promotes greater intervention and discussion between the participants.

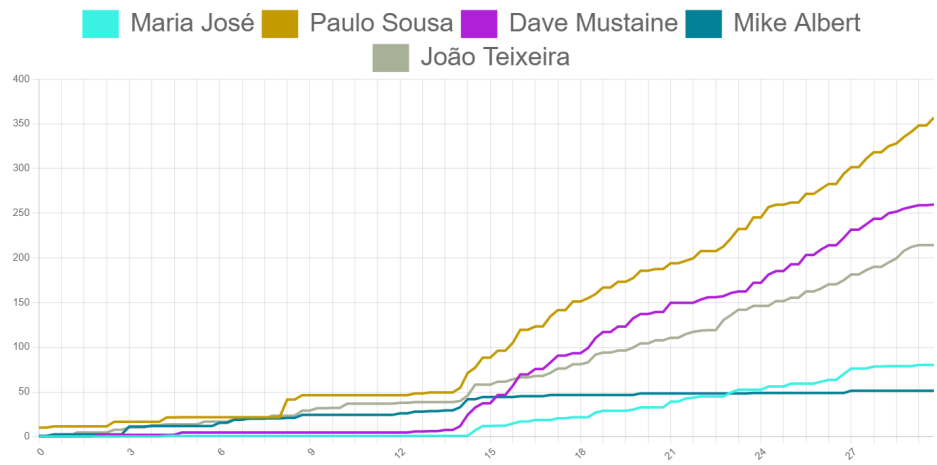


Figure 6.1: Panel from the dashboard presenting the individual speaking time during one meeting

6.1.2 Intervention Time vs Speaking Time

The system also makes it possible to compare speaking and intervention time, as illustrated in Figure 6.2.

The speaking time is obtained from analyzing all the audio files with my-voice-analysis and summing the detected speaking time.

The intervention time is the sum of the duration of all audio segments. It includes pauses during the speech, accidental noise, and the total speaking time itself.

Data is organized individually by participants and a meeting average is also provided for a better analysis.

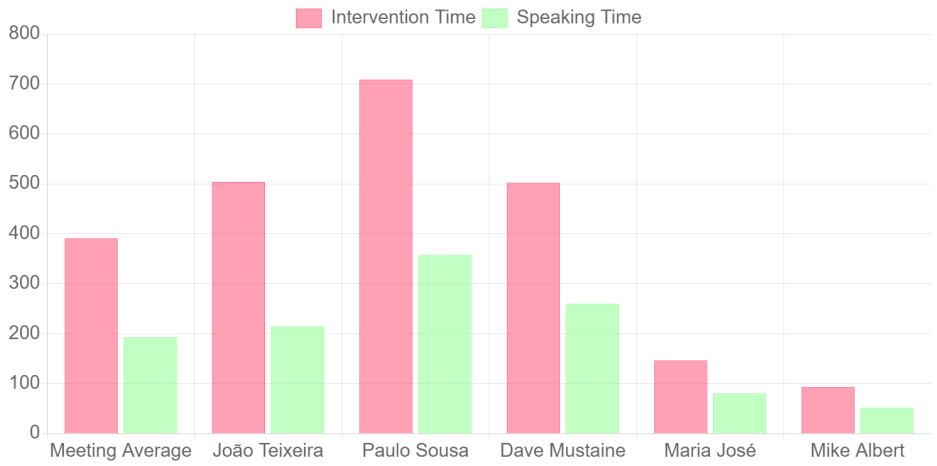


Figure 6.2: Intervention Time vs Speaking Time

This plot enables the coordinator to easily spot who talked more and less. For example, it can be easily spotted that Maria José and Mike Albert did not intervene as much compared to the other participants.

It is also possible to compare the intervention time to speaking time. High intervention times could mean that the system is detecting more noise than usual. In this case, the intervention time is around 50% higher in all of the participants. This was expected since all participants were in the same conditions, using the same microphone and in the same room. Although it is possible to see a slight difference between João Teixeira and Dave Mustaine, they both have the same intervention time but Dave Mustaine has a slightly higher speaking time.

To produce the values presented the system takes advantage of the knowledge base. In the listing 6.1, it is possible to see how this is achieved using a SPARQL query.

The presented query is divided into two parts: (1) Sum of the individual values, and (2) Calculation of the meeting average. To achieve this both gather all the audio from a specific meeting and the associated statistics. Since this information is organized by user, using the "SUM()" method will return the sum of the values for each participant.

For the calculation of the meeting average, an extra step is added. The information previously produced goes through the "AVG()" method that returns the average of the values introduced.

Listing 6.1: Intervention Time vs Speaking Time SPARQL query

```
#Individual Times
PREFIX meet: <http://www.semanticweb.org/joãoteixeira/ontologies/2021/4/meeting#>
SELECT ?user SUM(?speech) as ?speechsum SUM(?intervation) as ?intervationsum
FROM <meet_analyser> WHERE
{
    ?audio a meet:Audio;
        meet:hasMeeting <meet_id>;
        meet:hasStatistics ?stats;
        meet:hasUser ?user.

    ?stats a meet:Statistics;
        meet:speaking_duration ?speech;
        meet:original_duration ?intervation.
}
#Meeting Average
PREFIX meet: <http://www.semanticweb.org/joãoteixeira/ontologies/2021/4/meeting#>
SELECT AVG(?speechsum) as ?speechavg AVG(?intervationsum) ?intervationavg WHERE
{
    SELECT ?user SUM(?speech) as ?speechsum SUM(?intervation) as ?intervationsum
    FROM <meet_analyser> WHERE
    {
        ?audio a meet:Audio;
            meet:hasMeeting <meet_id>;
            meet:hasStatistics ?stats;
            meet:hasUser ?user.

        ?stats a meet:Statistics;
            meet:speaking_duration ?speech;
            meet:original_duration ?intervation.
    }
}
```

6.1.3 Multidimensional Analysis of the Meeting

The Statistics module of the system provides several parameters. To make it easier to understand them jointly, essential to understanding quickly the meeting, a multidimensional graphical representation was considered useful and added to the system. Figure 6.3 presents a visualization displaying them in a radar chart.

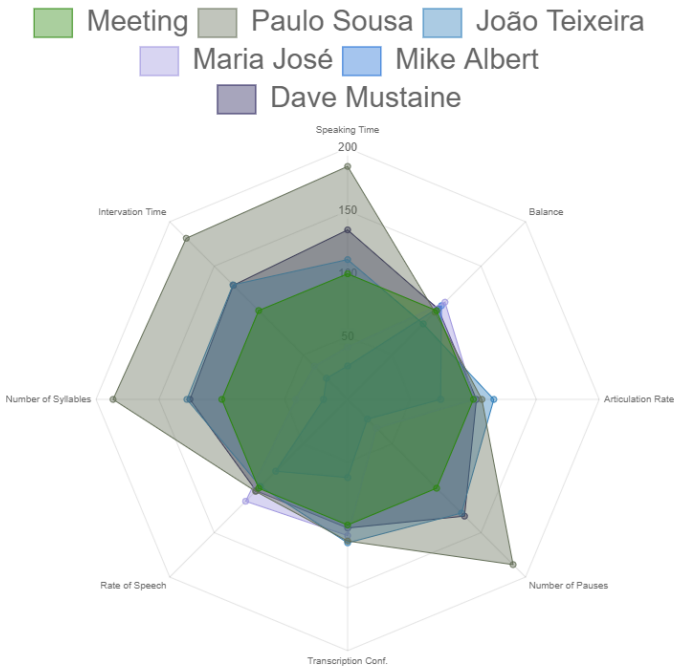


Figure 6.3: Multidimensional Analysis Chart

To produce these values it is used the query 6.2. Similar to the query presented before in the " Intervention Time vs Speaking Time" plot, it is divided into two parts.

The first one takes care of producing the individual values for each participant using the methods "SUM()" and "AVG()" depending on the variable. Using the speaking duration and the articulation rate as examples. The first one should be a sum, as it is important to know how much the participant talked in total. For the second one, it does not make sense to have the sum of the articulation rate, in this case, is used the average, giving the information of how fast a participant talks.

The second part takes care of producing the average value for the meeting. It is in everything similar to the first one, adding only a new level to query and using the "AVG()" method.

All these values are then normalized into percentages, to be better compared, using the formula presented below. As for the meeting average, it will be 100%.

$$userPercentage = \frac{userAverage}{meetingAverage} * 100$$

The amount of information presented by the visualization can be customized, allowing to display all the participants or only specific ones, making it easier to compare data between

them.

Listing 6.2: Multidimensional Analysis Query

```
#Individual Values
PREFIX meet: <http://www.semanticweb.org/joãoteixeira/ontologies/2021/4/meeting#>
SELECT ?user SUM(?speak) as ?speechsum SUM(?pauses) as ?pausessum
AVG(?balance) as ?balanceavg AVG(?artrate) as ?artrateavg
SUM(?ori) as ?orisum SUM(?ns) as ?nssum AVG(?ratespeech) as ?ratespeechavg

  FROM <meet_analyser> WHERE {
    ?audio a meet:Audio;
        meet:hasMeeting <essmdu2np7zj702m3amn>;
        meet:hasStatistics ?stats;
        meet:hasUser ?user.


      ?stats a meet:Statistics;
            meet:speaking_duration ?speak;
            meet:balance ?balance;
            meet:articulation_rate ?artrate;
            meet:number_of_pauses ?pauses;
            meet:rate_of_speech ?ratespeech;
            meet:number_of_syllables ?ns;
            meet:original_duration ?ori.

     FILTER (?speak>1).
}
#Meeting Average
PREFIX meet: <http://www.semanticweb.org/joãoteixeira/ontologies/2021/4/meeting#>

SELECT AVG(?speechsum) as ?speechsumavg AVG(?pausessum) as ?pausessumavg
AVG(?balanceavg ) as ?balanceavgavg AVG(?artrateavg ) as ?artrateavgavg
AVG(?orisum) as ?orisumavg AVG(?nssum ) as ?nssumavg
AVG(?ratespeechavg) as ?ratespeechavgavg

WHERE
{

    SELECT ?user SUM(?speak) as ?speechsum SUM(?pauses) as ?pausessum
    AVG(?balance) as ?balanceavg AVG(?artrate) as ?artrateavg
    SUM(?ori) as ?orisum SUM(?ns) as ?nssum AVG(?ratespeech) as ?ratespeechavg

    FROM <meet_analyser> WHERE
    {
        ?audio a meet:Audio;
            meet:hasMeeting <essmdu2np7zj702m3amn>;
            meet:hasStatistics ?stats;
            meet:hasUser ?user.


        ?stats a meet:Statistics;
               meet:speaking_duration ?speak;
```

```
                meet:balance ?balance;
                meet:articulation_rate ?artrate;
                meet:number_of_pauses ?pauses;
                meet:rate_of_speech ?ratespeech;
                meet:number_of_syllables ?ns;
                meet:original_duration ?ori.

        FILTER (?speak>1).
    }
}
```

6.1.4 Participants Mood

An important piece of information retrieved by the system is the mood of each participant. It provides some insight on user engagement throughout the meeting. This information is collected using the my-voice-analysis [45] package, and it can take the following values: "Reading", "Speaking passionately", and "Showing no emotion".

Figure 6.4 shows how the dashboard exposes the most common mood during the meeting for each one of the participants. The mood that showed to be more correlated to higher interventions and greater discussion was "Speaking passionately".

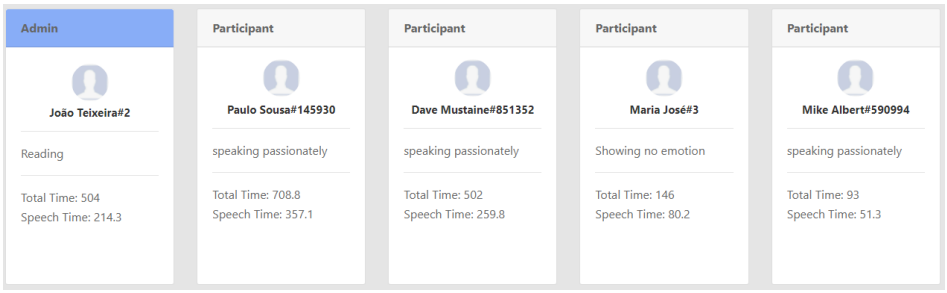


Figure 6.4: Panel from the dashboard presenting the information of mood of speech and total intervention and speech time

6.1.5 Transcription

To make possible further processing and analysis, and support the creation of meeting minutes the system provides a full transcription of the meeting. This information is visible in the dashboard as presented in Figure 6.5.

Each intervention presents the participant, date/time, and the actual audio transcript. Users can also listen to the intervention so they can effectively compare the transcript with the collected audio.

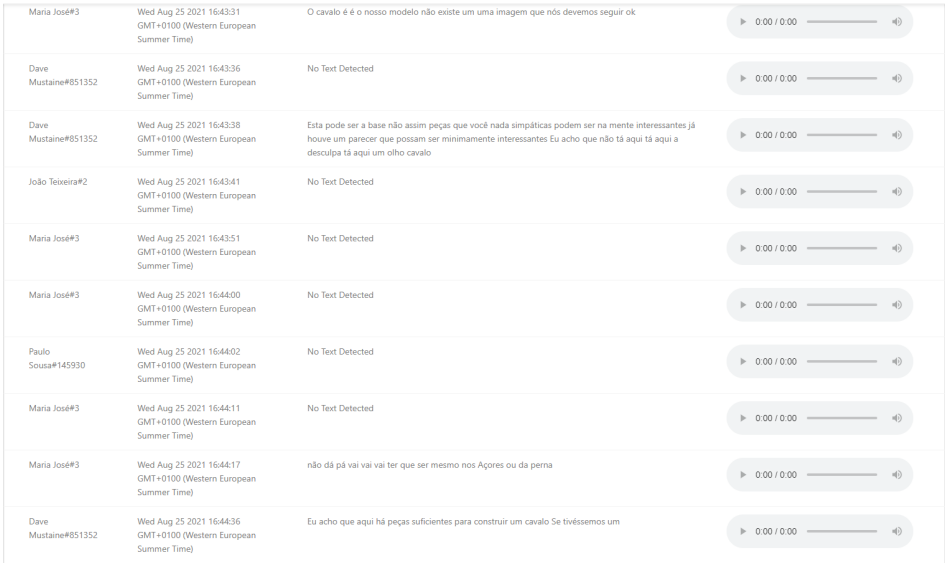


Figure 6.5: Meeting Transcription on the dashboard. From the left to the right it displays the name of the user, date/time, the recognized text and a button to play the original audio.

Performance depends on the speech recognition service used and on the quality of the audio gathered. In rare cases, some entries show "No Text Detected", which means that a conflict was found between the speech-to-text and the "Voice Analysis" modules. The speech-to-text module was not capable of finding any speech to transcribe, but the voice analysis package detected speech time. Since it is impossible to conclude which one is wrong the information will be always shown as it might be crucial.

6.1.6 PDF generation

Although the results presented are always available through the web interface, a PDF conversion is also available. This information is rendered into two different files: statistical data, and transcription data.

The statistical data is a replica of what is shown in the dashboard. It works by converting HTML elements into PDF. This implementation has some drawbacks. It has low performance and is not possible to select/copy any text from it.

Because of these limitations for the transcription data, a PDF file is created and the information is written into it. In Figure 6.6 it is possible to see how this file is rendered. In this case, all the entries marked with "No Text Detected" are filtered since they do not represent significant information in this context.

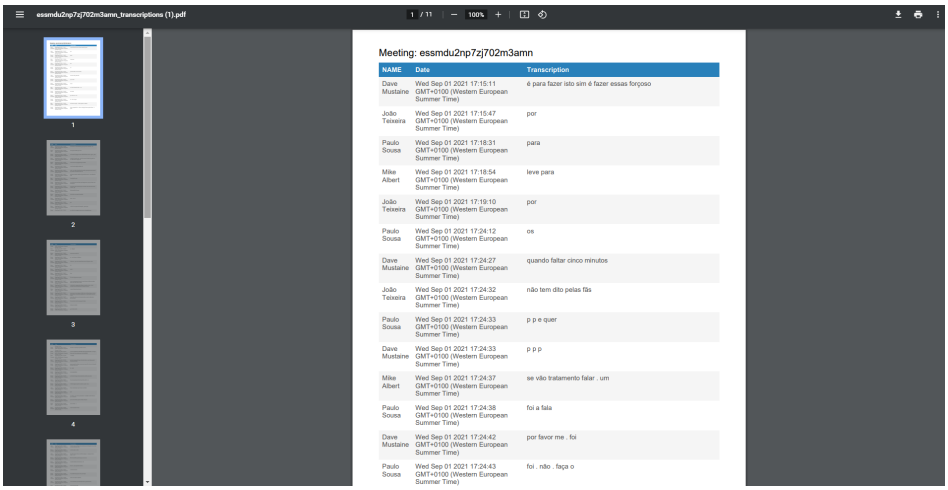


Figure 6.6: Transcription PDF file

6.2 Multi-meeting statistics

In this chapter were already shown some of the capabilities of the system to help in improving meetings with the rich and easy-to-use dashboard.

This section will focus on processing a set of meetings. Producing a set of statistics divided into two types, global statistics and user statistics.

The first one presents averages from all the participants from all the meetings that the system processed.

The second one is similar to the first but in this case, presents the averages from all the meetings but to a particular user.

6.2.1 Global Averages

Taking into consideration all the values extracted by the "Voice Analysis" module the system does a global average from all the meeting and their participants. This can give a better understanding of how usually a participant behaves during a meeting.

The listing 6.3 bellow shows how this is done using the knowledge base. The SPARQL query is similar to the one used in the "Multidimensional Analysis" for the meeting average. Gathering all the values for each participant, but this time with no meeting restriction. The query is composed of three steps.

The first one gathers all the statistical values from all speech processed by the system from all the meetings and their participants (Figure 6.7).

The second step calculates the averages for each meeting (Figure 6.8).

Lastly, to get the global averages from all the processed meetings it is done the average of the values gathered in the step before (Figure 6.9).

Listing 6.3: Global averages extracted form all the meetings

```
PREFIX meet: <http://www.semanticweb.org/joãoteixeira/ontologies/2021/4/meeting#>
SELECT AVG(?speechsumavg) as ?SpeakDuration AVG(?pausessumavg) as ?NumberPauses
AVG(?balanceavgavg) as ?Balance AVG(?artrateavgavg ) as ?ArticulationRate
AVG(?orisumavg ) as ?OriginalDuration AVG(?nssumavg ) as ?NumberSyllables
AVG(?ratespeechavgavg ) as ?RateOfSpeech
WHERE
{

SELECT ?meet AVG(?speechsum) as ?speechsumavg AVG(?pausessum) as ?pausessumavg
AVG(?balanceavg ) as ?balanceavgavg AVG(?artrateavg ) as ?artrateavgavg
AVG(?orisum) as ?orisumavg AVG(?nssum ) as ?nssumavg
AVG(?ratespeechavg) as ?ratespeechavgavg
WHERE
{

    SELECT ?meet ?user SUM(?speak) as ?speechsum SUM(?pauses) as ?pausessum
    AVG(?balance) as ?balanceavg AVG(?artrate) as ?artrateavg
    SUM(?ori) as ?orisum SUM(?ns) as ?nssum AVG(?ratespeech) as ?ratespeechavg
    FROM <meet_analyser> WHERE
    {
        ?audio a meet:Audio;
            meet:hasMeeting ?meet;
            meet:hasStatistics ?stats;
            meet:hasUser ?user.

        ?stats a meet:Statistics;
                meet:speaking_duration ?speak;
                meet:balance ?balance;
                meet:articulation_rate ?artrate;
                meet:number_of_pauses ?pauses;
                meet:rate_of_speech ?ratespeech;
                meet:number_of_syllables ?ns;
                meet:original_duration ?ori.

        FILTER (?speak>0).
    }
}
}
```

meet	user	speechsum	pausessum	balanceavg	artrateavg	orisum	nssum	ratespeechavg
hjbu6gipyzvn9ittncu5	3	96.3	41	0.48	3.83	194	370	1.77
iadnb8vu1112ui3axrz7	2	20.3	6	0.61	3.4	31.6	73	2.16
t507jq95zlo6qpiu4cx1	3	33	19	0.43	3.93	78	133	1.69
tmr2ucebatrez53ayxgd	590994	7.8	4	0.65	3.5	12	27	2
1a4mdu5hrego464xnkql	2	59.6	21	0.55	4.25	83.5	278	2.25
7p0npex9w93b3sw32kx1	2	91.1	0	0.19	1.61	133.2	157	0.54
t507jq95zlo6qpiu4cx1	851352	25.4	15	0.66	4.29	40	108	2.86
nhlb4e4f5edjz17wzy8f	2	249.7	138	0.5	3.82	486	972	1.92
f69eoizk73syfnjmcx5x	851352	112.1	35	0.57	3.8	187.9	456	2.2
9gidefcuhnbp67xgcgol	3	52.4	18	0.8	5	67.3	249	4
h0cv9shdfeqf3hf9x238	2	54.3	52	0.25	4.4	379	232	1.06
nhlb4e4f5edjz17wzy8f	590994	3.3	0	0.37	2	9	8	0.87
tmr2ucebatrez53ayxgd	3	400.5	141	0.45	3.8	872	1668	1.7
tmr2ucebatrez53ayxgd	145930	424	163	0.5	4.23	805	1903	2.01
hjbu6gipyzvn9ittncu5	851352	68.7	37	0.48	3.44	147	244	1.51
2su47px5rtfs2wnd0pjv	145930	247	223	0.43	2.83	631	700	1.16
essmdu2np7zj702m3amn	851352	259.8	154	0.51	3.71	502	927	1.81
2su47px5rtfs2wnd0pjv	590994	21.9	56	0.23	1.7	145	30	0.45

Figure 6.7: Results for the first step of the query 6.3

meet	speechsumavg	pausessumavg	balanceavgavg	artrateavgavg	orisumavg	nssumavg	ratespeechavgavg
nhlb4e4f5edjz17wzy8f	224.22	112.2	0.440525542667704	3.435022847566709	480.2	865.8	1.519590341609117
1a4mdu5hrego464xnkql	59.6	21	0.55	4.25	83.5	278	2.25
t507jq95zlo6qpiu4cx1	18.88	13.4	0.442761904761905	3.557142857142857	42.6	70.2	1.657428571428571
yx3o40p2ynanziqo981n	19.05	5.5	0.613461538461538	4.125	29.2	80	2.7875
essmdu2np7zj702m3amn	192.54	117.2	0.504585063848913	3.626473487516414	390.76	739.6	1.757758194994202
hjbu6gipyzvn9ittncu5	65.6	32.8	0.549330794208843	3.349406442089369	134	249.2	1.791695034621864
h0cv9shdfeqf3hf9x238	137.35	105.25	0.388910422316931	3.802077683068578	375	465.5	1.294670670282836
f69eoizk73syfnjmcx5x	87.233333333333333	23.666666666666667	0.544279411764706	4.076149956838081	150.1	372.666666666666667	2.26520224441978
tmr2ucebatrez53ayxgd	281.08	117	0.539372789737496	3.817860627390039	540.8	1161	1.967806724865548
9gidefcuhnbp67xgcgol	52.566666666666667	18	0.683333333333333	4.5	68.5	249.333333333333333	3.383333333333333
iadnb8vu1112ui3axrz7	27.833333333333333	6.666666666666667	0.565982905982906	3.787179487179487	48	113	2.128395368072787
7p0npex9w93b3sw32kx1	91.1	0	0.191791044776119	1.611940298507463	133.2	157	0.538059701492537
2su47px5rtfs2wnd0pjv	198.4	174.2	0.387012820512821	3.211196581196581	545.8	648.4	1.291068376068376

Figure 6.8: Results for the second step of the query 6.3

SpeakDuration	NumberPauses	Balance	ArticulationRate	OriginalDuration	NumberSyllables	RateOfSpeech
111.96	57.45	0.49	3.63	232.44	419.21	1.89

Figure 6.9: Results for the third and last step of the query 6.3

6.2.2 User statistics

In the same way that the "Global Statistics" are produced it is possible to do the same for a single user.

To calculate the individual statistics for a particular user the method used is similar to what was done on the listing 6.3 presented in the previous section.

In the first part of the query instead of returning the values for each participant only return for the specified one (Figure 6.10). This is done by changing the line "meet:hasUser ?user" into "meet:hasUser <USERID>", where "USERID" is the unique ID of the user.

Since there are no multiple participants this removes one step from the query compared to the listing 6.3. In this case the second step, in charge of calculating the averages for each meeting., is not needed.

The last step is the same as in the previous section. It calculates the average from the values previously extracted (Figure 6.11).

meet	speechsum	pausessum	balanceavg	artrateavg	orisum	nssum	ratespeechavg
nhlb4e4f5edjz17wzy8f	247.7	98	0.42	4.03	563	1052	1.65
t507jq95zlo6qpiu4cx1	24.3	18	0.43	2.4	61	64	1.02
essmdu2np7zj702m3amn	357.1	218	0.5	3.87	708.8	1379	1.82
hjbu6gipyzvn9ittncu5	82.9	37	0.43	3.74	186	329	1.59
h0cv9shdfeqf3hf9x238	138.4	108	0.42	3.34	350	416	1.26
tmr2ucebatrez53ayxgd	424	163	0.5	4.23	805	1903	2.01
2su47px5rtfs2wnd0pjv	247	223	0.43	2.83	631	700	1.16

Figure 6.10: Query result with user values by meeting

SpeakDuration	NumberPauses	Balance	ArticulationRate	OriginalDuration	NumberSyllables	RateOfSpeech
217.34	123.57	0.45	3.49	472.11	834.71	1.5

Figure 6.11: Query result for user average values

The values gathered can then be compared to what was obtained in the "Global Averages" section.

These values can be normalized into a percentage using the equation presented below. Making it possible to produce values in a similar way to what was done in Section 6.1.3. Being the "Global Averages" 100% as on the "Meeting Averages". These can also be introduced into a radar graph as shown in Figure 6.12

$$userPercentage = \frac{userAverage}{globalAverage} * 100$$

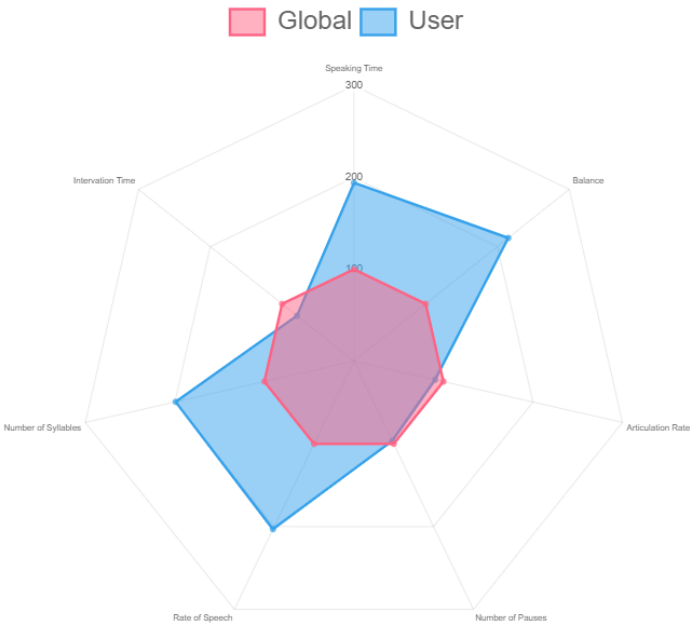


Figure 6.12: Chart presenting global average vs individual values

With a simple adjustment to this equation, it is possible to calculate the relative variation percentage-wise.

$$relativeVariation = userPercentage - 100$$

6.3 Alerts System

The Alerts is the way that the system has to interact in real-time with the coordinator or the participants. Although the dashboard can be rich and is built in a way that the participants can easily search for the information needed it requires the user to shift focus. This can have a negative impact on the meeting, doing the opposite of what the platform tries to achieve.

The system has two types of alerts as explained before: (1) Performed by the meeting coordinator and (2) Automated Alerts.

This section is focused on the Automated Alerts since the other type was already explained in section 5.2.2.4.

The system has a set of automated alerts, described below, that can help the participants through the meeting. They can be activated/disabled and configured through the "Alert Configuration" interface only available to the coordinator.

It is also important to highlight that no post-processing is done by the Alerts System. This section besides presenting the alerts that the system is capable of producing, it is also explained how this is done only by querying the knowledge base.

6.3.1 Transcription confidence

The quality of transcriptions affects the performance of all system processing modules. As a result, is important to be notified if quality degrades.

The System implements an alert that is activated whenever average confidence is below a user-defined threshold.

An example of this alert is shown in Figure 6.13.

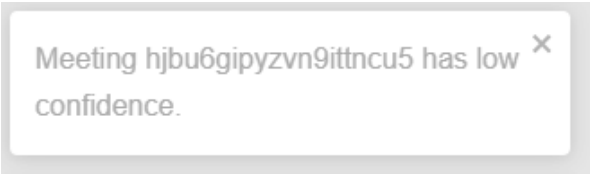


Figure 6.13: Alert toast showing that the meeting average transcription confidence is low

The system periodically queries the knowledge base to find the current average value of the confidence as shown in the listing 6.4. Given a meeting ID the knowledge base gathers all the audio files entries and corresponding transcription confidence. A filter is used to remove all the entries where speech was not detected.

Lastly, the average is calculated and then compared to the value defined by the meeting coordinator, in this example "0.4". If the value is lower than the threshold the knowledge base will return "1" and the system will send the alert to all the meeting participants.

Listing 6.4: "Average Transcription Confidence"

```
PREFIX meet: <http://www.semanticweb.org/joãoteixeira/ontologies/2021/4/meeting#>
SELECT IF(AVG(?conf) < 0.4, 1, 0) as ?Average_Conf FROM <meet_analyser>

WHERE {

    ?audio a meet:Audio;
        meet:hasMeeting <meet_id>;
        meet:hasTranscription ?transcription .

    ?transcription a meet:Transcription;
        meet:confidence ?conf .

    FILTER ( ?conf > 0 ) .
}
```

6.3.2 Speaking Duration Limit

This alert is shown when a participant exceeds the intervention time limit (in seconds), defined by the meeting coordinator in the "Alert Configuration" interface. The alert toast is illustrated in Figure 6.14.

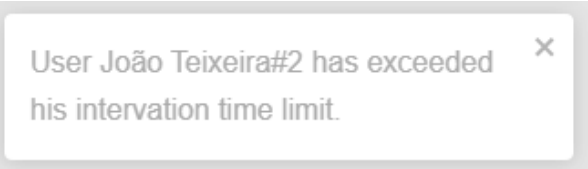


Figure 6.14: Alert toast for participants that exceed the intervention time limit

The knowledge base is periodically queried for this information as presented below in the listing 6.5. Given a meeting ID, the knowledge base returns all the audio files entries, their user, and the speaking duration statistic. These are then grouped by user, and the speaking duration is summed and compared to the values defined as the threshold by the meeting coordinator, in this example "120".

The knowledge base only returns the user IDs that exceed that value and the system proceeds to send the alert to the particular users and the meeting coordinator.

Listing 6.5: Global averages extracted form all the meetings

```
PREFIX meet: <http://www.semanticweb.org/joãoteixeira/ontologies/2021/4/meeting#>
SELECT ?user FROM <meet_analyser> where {
    ?audio a meet:Audio;
        meet:hasMeeting <meet_id>;
        meet:hasStatistics ?stats;
        meet:hasUser ?user.

    ?stats a meet:Statistics;
        meet:speaking_duration ?speech.

} GROUP BY ?user HAVING (SUM(?speech) > 120)
```

6.3.3 Speech Duration departing from average

To help detect and fight the non-participating and dominant behaviors, essential to improve meetings as mentioned in chapter 2, a third and very important alert type was implemented.

This alert is shown when a participant is intervening substantially more or less than average (Figure 6.15).

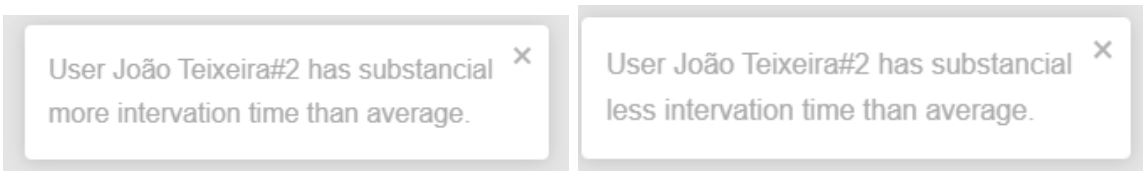


Figure 6.15: Alert toasts for substantial more/less intervention time than average.

First, the meeting coordinator defines the upper and lower limit allowed to speak above and below average, in this example "45%".

Then using the formulas presented below, it's possible to compare the individual speaking duration to the meeting average.

$$SUM(indivudualSpeakingDuration) > meetingAverage + meetingAverage * 0.45$$

$$SUM(indivudualSpeakingDuration) < meetingAverage - meetingAverage * 0.45$$

The knowledge base is periodically queried for this information as presented below in the listing 6.5. The query returns the users that crossed the threshold for high or low participation in the meeting. This query is more complex than previously demonstrated queries having four levels.

The first one, given a meeting "ID," gathers all the audio files, their users, and the speaking duration statistic. With this information, the sum of individual speaking duration is calculated (Figure 6.16a).

The second level calculates the total meeting average using the individual sum obtained before (Figure 6.16b).

The third one does the same as the first one but this time it also returns the meeting average alongside the user id and the sum of the speaking duration (Figure 6.16c).

The last one filters the values obtained using the formulas presented before and also compares the individual speaking duration to the average if it is higher the last column will be set to "1" otherwise it will be set to "0". This way it is possible to differentiate if the user is intervening above or below average and send the correspondent alert (Figure 6.16d).

```
PREFIX meet: <http://www.semanticweb.org/joãoteixeira/ontologies/2021/4/meeting#>
SELECT ?user IF(?usersum > ?averagetotal, 1, 0) WHERE
{
FILTER ( ?usersum > (?averagetotal + ?averagetotal * 0.45) OR
?usersum < (?averagetotal - ?averagetotal * 0.45) )
    {
    SELECT ?user SUM(?speech) as ?usersum ?averagetotal FROM <meet_analyser> WHERE
    {
        ?audio a meet:Audio;
            meet:hasMeeting <meet_id>;
            meet:hasStatistics ?stats;
            meet:hasUser ?user.

            ?stats a meet:Statistics;
                meet:speaking_duration ?speech.
        {
            SELECT AVG(?sumofSpeech) as ?averagetotal WHERE
            {
                SELECT ?user SUM(?speech) as ?sumofSpeech FROM <meet_analyser> WHERE
                {
                    ?audio a meet:Audio;
                        meet:hasMeeting <meet_id>;
                        meet:hasStatistics ?stats;
                        meet:hasUser ?user.
```

```
                ?stats a meet:Statistics;
                    meet:speaking_duration ?speech.
            }
        }
    }
    }
}
```

user	sumofSpeech
2	214.3
590994	51.3
851352	259.8
3	80.2
145930	357.1

(a) Query result with the sum of speech time for all the participants in one specific meeting

averagetotal
192.54

(b) Query result with the meeting average taking in consideration the values produced in Figure 6.16a

user	usersum	averagetotal
3	80.2	192.54
590994	51.3	192.54
2	214.3	192.54
851352	259.8	192.54
145930	357.1	192.54

(c) Query result with the sum of speech time for all the participants in one specific meeting and the meeting average.

user	higher
145930	1
3	0
590994	0

(d) Query result with the participants that intervene substantially more or less than average.

Figure 6.16: Group of query results produces from the Query 6.3.3

Chapter 7

Conclusions

> If we have a clear agenda in advance and we are fully present and fully contributing, the meetings do go much faster.
>
> *(Arianna Huffington)*

This chapter concludes the dissertation with an overall summary of the work done. It starts by enumerating the main phases of development and their outcome. Followed by the presentation of the main results extracted from the work. Lastly, to finish this chapter, some topics are addressed related to future works that can be performed on top of the one described in this dissertation.

7.1 Work summary

Before the development of the proof-of-concept system, the work started by studying and understanding the general concept of meetings. It was followed by the search and analysis of technologies and related work in this subject.

After this the development of the proof-of-concept system started and had the following phases:

- **Requirements**, the first step of development was a creation of a list of requirements to be fulfilled and order them by priority. These requirements are presented in section 3.3.
- **Search of the meeting platform**, since the beginning meetings in person were impossible to do. The system shifted to a 100% online format meaning that a platform to hold the meetings was needed. This phase was completed with the selection of Jitsi [41];
- **Architecture design**, the decoupled architecture focused on real-time streaming of data, provided the flexibility that facilitated the development and maintenance of modules;
- **Creation of the first modules** with the implementation of the audio recording and speech-to-text modules;

- **Creation of semantic knowledge database**, although time-consuming showed to be important later, as it gave a better tool to analyze data and produce information, and also flexibility to be further expanded;
- **Implementation of more modules** with the implementation of the voice analysis, statistics, dashboard and alert modules;
- **Processing of pre-recorded meetings** was the last phase of development, at this point, the system was ready to be tested, and using several pre-recorded meetings gave the system access to a high number of data in a short time.

7.2 Main results

The proof-of-concept system showed to be already capable of some degree of processing. It can "listen" to meetings, detect speech and speaker and then process the audio. Using different tools, such as speech-to-text or voice analysis, produces useful information regarding the meeting and participants.

The alert system also demonstrated potential, giving the system the ability to intervene in the meeting and help in its development.

Although this showed some interesting results it should not be seen as the only results of the system. Since the beginning of development, one of the main focuses was to create a system that could be easily reused by other researchers and could be expandable and improved without the need for high restructuring and redesign.

This was achieved first by the architecture design and implementation. Using a decoupled architecture focused on real-time streaming of data, provided flexibility to add, remove or modify any module without the need to change the overall system.

Lastly by the utilization of a semantic database and the creation of the respective ontology, removing any storage barrier that could exist. The ontology can, without a lot of effort, be expandable to fulfill the needs of new modules.

7.3 Future work

As the presented work is both a first step and an initial proof-of-concept, the future work is rich and covers distinct lines of research, the most relevant being:

- Generation of meetings summaries;
- Evaluation at the end of each meeting, scoring the current meeting, to connect the data collected to good or bad meeting experience;
- Chat-bot able to answer user questions, complement the dashboard, and make access to information more natural. The bot can evolve to participate in the meeting, intervening to make the meeting more efficient;
- Experimenting with others Speech-to-Text systems;
- Exploration of Machine learning. With all the data collected, it can be used for alerts, the chat-bot or even being the coordinator of the meeting. The applications are endless.

Bibliography

[1] Apache. Apache kafka. https://kafka.apache.org/, 2021. (accessed 27/07/2021).

[2] Michel Assayag, Jonathan Huang, Jonathan Mamou, Oren Pereg, Saurav Sahay, Oren Shamir, Georg Stemmer, and Moshe Wasserblat. Meeting assistant application. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[3] Amazon AWS. Pub/sub messaging. https://aws.amazon.com/pt/pub-sub-messaging/, 2021. (accessed 27/07/2021).

[4] Paul Boersma and David Weenink. Praat: doing phonetics by computer. https://www.fon.hum.uva.nl/praat/, 2021. (accessed 28/07/2021).

[5] BOOQED. Minutes (wasted) of meeting: 50 shocking meeting statistics. https://www.booqed.com/blog/minutes-wasted-of-meeting-50-shocking-meeting-statistics, 2021. (accessed 21/09/2021).

[6] Cambridge Semantics. Owl 101. https://www.cambridgesemantics.com/blog/semantic-university/learn-owl-rdfs/owl-101/, 2021. (accessed 31/08/2021).

[7] Dale Carnegie. *A arte de comunicar com sucesso*. Objectiva, 2021.

[8] Tathagata Chakraborti, Kshitij P. Fadnis, Kartik Talamadupula, Mishal Dholakia, Biplav Srivastava, Jeffrey O. Kephart, and Rachel K. E. Bellamy. Visualizations for an explainable planning agent, 2018.

[9] Senthil Chandrasegaran, Chris Bryan, Hidekazu Shidara, Tung-Yen Chuang, and Kwan-Liu Ma. Talktraces: Real-time capture and visualization of verbal content in meetings. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

[10] Colorlib. Cooladmin bootstrap 4.1 admin dashboard template. https://github.com/puikinsh/CoolAdmin, 2021. (accessed 31/08/2021).

[11] Colorlib. Cooladmin bootstrap 4.1 admin dashboard template. https://colorlib.com/polygon/cooladmin/index.html, 2021. (accessed 31/08/2021).

[12] Chart.js contributors. Chart.js. https://www.chartjs.org/, 2021. (accessed 31/08/2021).

[13] Chart.js contributors. react-chartjs-2. https://www.npmjs.com/package/react-chartjs-2, 2021. (accessed 31/08/2021).

[14] D3.js contributors. D3.js. https://d3js.org/, 2021. (accessed 31/08/2021).

[15] FFmpeg contributors. Ffmpeg. https://www.ffmpeg.org/, 2021. (accessed 31/08/2021).

[16] Jitsi contributors. Jitsi. https://github.com/jitsi/lib-jitsi-meet, 2021. (accessed 28/07/2021).

[17] Jitsi contributors. The jitsi handbook. https://github.com/jitsi/handbook/blob/master/docs/architecture.md, 2021. (accessed 28/07/2021).

[18] Jitsi contributors. jitsi-meet. https://github.com/jitsi/jitsi-meet, 2021. (accessed 28/07/2021).

[19] MDN contributors. The websocket api (websockets). https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API, 2021. (accessed 27/07/2021).

[20] RecordRTC contributors. Recordrtc. https://recordrtc.org/, 2021. (accessed 28/07/2021).

[21] Socket.IO contributors. Socket.io. https://socket.io/docs/v4, 2021. (accessed 27/07/2021).

[22] WebRTC contributors. Webrtc. https://webrtc.org/, 2021. (accessed 28/07/2021).

[23] Alan Cooper, Robert Reimann, and David Cronin. *About Face3: The Essentials of Interaction Design*. Willey Publishing, 2007.

[24] J Cuomo. Javascript everywhere and the three amigos (into the wild blue yonder!). *IBM,-08-142015, Saatavissa: https://www. ibm. com/developerworks/community/blogs/gcuomo/entry/javascript_everywhere_and_the_three_amigos*, 2013.

[25] DataONE. Protégé. https://old.dataone.org/software-tools/protege, 2021. (accessed 31/08/2021).

[26] Nivja H De Jong and Ton Wempe. Praat script to detect syllable nuclei and measure speech rate automatically. *Behavior research methods*, 41(2):385–390, 2009.

[27] Brian Dean. Zoom user stats: How many people use Zoom in 2021? https://backlinko.com/zoom-users, 2021. (accessed 22/04/2021).

[28] Brian Dean. Zoom user stats: How many people use zoom in 2021? https://backlinko.com/zoom-users, 2021. (accessed 21/09/2021).

[29] Ketan Doshi. Audio deep learning made simple: Automatic speech recognition (asr), how it works. https://bit.ly/3nnx5Xc, 2021. (accessed 28/07/2021).

[30] Facebook. Facebook login. https://developers.facebook.com/docs/facebook-login/, 2021. (accessed 28/07/2021).

[31] Facebook. React. https://reactjs.org/, 2021. (accessed 25/08/2021).

[32] OpenJS Foundation. Express. https://expressjs.com/, 2021. (accessed 28/07/2021).

[33] Python Software Foundation. Python. https://www.python.org/, 2021. (accessed 28/07/2021).

[34] GOOGLE. Speech-to-text. https://cloud.google.com/speech-to-text, 2021. (accessed 28/07/2021).

[35] Carlos Gussenhoven. Intonation and interpretation: phonetics and phonology. In *Speech Prosody 2002, International Conference*, 2002.

[36] IBM. Ibm cloud docs languages and models. https://cloud.ibm.com/docs/speech-to-text?topic=speech-to-text-models, 2021. (accessed 28/07/2021).

[37] IBM. Message brokers. https://www.ibm.com/cloud/learn/message-brokers, 2021. (accessed 27/07/2021).

[38] IBM. Watson speech to text. https://www.ibm.com/cloud/watson-speech-to-text, 2021. (accessed 28/07/2021).

[39] SRI International. 75 years of innovation: Calo (cognitive assistant that learns and organizes). https://bit.ly/3njgKDd, 2021. (accessed 28/07/2021).

[40] Y. Jadoul, B. Thompson, and B. de Boer. Introducing parselmouth: A python interface to praat. https://www.sciencedirect.com/science/article/pii/S0095447017301389?via%3Dihub, 2021. (accessed 28/07/2021).

[41] Jitsi. Jitsi. https://jitsi.org/, 2021. (accessed 28/07/2021).

[42] Joyent. Node.js. https://nodejs.org/en/, 2021. (accessed 28/07/2021).

[43] Simone Kauffeld and Nale Lehmann-Willenbrock. Meetings matter: Effects of team meetings on team and organizational success. *Small group research*, 43(2):130–158, 2012.

[44] Been Kim and Cynthia Rudin. Learning about meetings. *Data mining and knowledge discovery*, 28(5):1134–1157, 2014.

[45] Sab-AI Lab. my-voice-analysis. https://github.com/Shahabks/my-voice-analysis, 2021. (accessed 28/07/2021).

[46] Sab-AI Lab. Sab-ai lab. https://shahabks.github.io/Sab-AI-Lab/, 2021. (accessed 28/07/2021).

[47] Hui Liu, Xin Wang, Yuheng Wei, Wei Shao, Jonathan Liono, Flora D Salim, Bo Deng, and Junzhao Du. Prometheus: An intelligent mobile voice meeting minutes system. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 392–401, 2018.

[48] Iva Marinova. 28 need-to-know remote work statistics of 2021. https://review42.com/resources/remote-work-statistics/, 2021. (accessed 28/07/2021).

[49] MODALITY. 30 virtual meeting statistics you should know. https://www.modalitysystems.com/hub/blog/virtual-meeting-statistics, 2021. (accessed 21/09/2021).

[50] Nelson Morgan, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Adam Janin, Thilo Pfau, Elizabeth Shriberg, and Andreas Stolcke. The meeting project at icsi. In *Proceedings of the first international conference on human language technology research*, 2001.

[51] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[52] ontotext. What is rdf? https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf/, 2021. (accessed 31/08/2021).

[53] ontotext. What is sparql? https://www.ontotext.com/knowledgehub/fundamentals/what-is-sparql/, 2021. (accessed 31/08/2021).

[54] OpenLink. Virtuoso. https://virtuoso.openlinksw.com/, 2021. (accessed 31/08/2021).

[55] Oracle. What is a relational database (rdbms)? https://www.oracle.com/database/what-is-a-relational-database/, 2021. (accessed 31/08/2021).

[56] Oracle. What is an object? https://docs.oracle.com/javase/tutorial/java/concepts/object.html, 2021. (accessed 31/08/2021).

[57] Steven G Rogelberg, Cliff Scott, and John Kello. The science and fiction of meetings. *MIT Sloan management review*, 48(2):18–21, 2007.

[58] Gokhan Tur, Andreas Stolcke, Lynn Voss, John Dowding, Benoît Favre, Raquel Fernández, Matthew Frampton, Michael Frandsen, Clint Frederickson, Martin Graciarena, et al. The CALO meeting speech recognition and understanding system. In *2008 IEEE Spoken Language Technology Workshop*, pages 69–72. IEEE, 2008.

[59] Gokhan Tur, Andreas Stolcke, Lynn Voss, Stanley Peters, Dilek Hakkani-Tur, John Dowding, Benoit Favre, Raquel Fernández, Matthew Frampton, Mike Frandsen, et al. The CALO meeting assistant system. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1601–1611, 2010.

[60] Stanford University. Protégé. https://protege.stanford.edu/, 2021. (accessed 31/08/2021).

[61] L. Lynn Voss and Patrick Ehlen. The CALO meeting assistant. In *HLT-NAACL (Demonstrations)*, pages 17–18, 2007.

[62] W3C. Rdf. https://www.w3.org/RDF/, 2021. (accessed 31/08/2021).

[63] W3C. Semantic web. https://www.w3.org/standards/semanticweb/, 2021. (accessed 31/08/2021).

[64] W3C. Virtuoso. https://www.w3.org/2001/sw/wiki/OpenLink_Virtuoso, 2021. (accessed 31/08/2021).

[65] W3C OWL Working Group. OWL. https://www.w3.org/OWL/, 2021. (accessed 31/08/2021).

[66] Silke M Witt and Steve J Young. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech communication*, 30(2-3):95–108, 2000.

[67] Zoom. Zoom. https://zoom.us/, 2021. (accessed 28/07/2021).

[68] Zoom. Zoom stream. https://support.zoom.us/hc/en-us/articles/115001777826-Live-streaming-meetings-or-webinars-using-a-custom-service, 2021. (accessed 28/07/2021).