universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Extracting facts from news videos. An approach to structuring information for automated fact-checking.

verfasst von | submitted by

Angelika Eva Kapeller BA BA BSc MA

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 935
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Medieninformatik
Betreut von | Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. Wolfgang Klas

Acknowledgements

First, I would like to thank my supervisor, Univ.-Prof. Dipl.-Ing. Dr. Wolfgang Klas, for guiding me through this master's thesis, from choosing a topic to implementing the prototype and completing the written work. His support and advice were very helpful whenever I faced difficulties, and his guidance was inspiring.

I also want to thank Dipl.-Ing. Dr. Peter Kalchgruber for encouraging me to explore a master's thesis topic in fact-checking during earlier courses in the master's program.

Furthermore, I am grateful to my fellow students whom I met during my time at the University of Vienna. I could not have completed my studies without them, and some have become lifelong friends. Special thanks to Benni, Michi, and Sandra.

Finally, I am grateful to my family for their support and encouragement throughout my studies, which helped me complete this thesis.

Abstract

News videos are an important source of information. However, their rapid dissemination via digital platforms is prone to the spread of misinformation and therefore requires reliable automated fact-checking methods. Unlike text-based news, videos combine spoken language, visual overlays, and metadata, making them difficult to analyse automatically. This master's thesis presents a system for extracting structured facts from news videos to support fact comparison and verification applications.

The thesis reviews related work in automated fact-checking and information extraction, with a special focus on video-based approaches. Building on this foundation, the proposed system integrates various technologies, including Microsoft Azure Video Indexer for video and audio analysis, and Natural Language Processing techniques such as Named Entity Recognition and extractive text summarisation. The core of the approach is an entity-based fact extraction system that identifies persons, organisations, and events and organises their properties using a schema based on schema.org standards.

A prototype was implemented as a modular fact extraction pipeline. Users can upload news videos, process them through the pipeline, and obtain structured fact reports in machine-readable format. The system integrates cloud-based and local NLP components to ensure robust fact extraction and normalisation.

The evaluation used a set of 31 BBC news videos. Results show that the system reliably extracts and structures facts about persons and events, with high accuracy in property assignment. However, the detection and description of organisation entities proved more challenging. Common challenges identified include misspelt names, vague or overly broad property values, and difficulties extracting precise temporal information.

In summary, this thesis demonstrates that automated fact extraction from news videos is both technically feasible and useful for structuring unstructured news video content. The prototype's results highlight areas for further improvement, especially in handling organisation entities, normalising property values, and extracting temporal data. Future work could extend the system to cover more entity types, improve value validation, and support additional languages and news sources.

Kurzfassung

Nachrichtenvideos sind eine wichtige Informationsquelle. Ihre schnelle Verbreitung über digitale Plattformen birgt jedoch das Risiko, Fehlinformationen zu verbreiten, und erfordert daher zuverlässige Methoden automatisierter Faktenchecks. Im Unterschied zu textbasierten Nachrichten vereinen Videos gesprochene Sprache, visuelle Elemente und Metadaten, was eine automatische Analyse erschwert. Diese Masterarbeit präsentiert einen Ansatz zur Extraktion strukturierter Fakten aus Nachrichtenvideos, die in weiterer Folge von Faktencheck-Anwendungen verarbeitet werden können.

Die Arbeit gibt einen Überblick über den Stand der Forschung im Bereich des automatisierten Faktenchecken sowie der Extraktion von Informationen. Ein besonderer Fokus liegt dabei auf videobasierten Ansätzen. Der entwickelte Prototyp integriert verschiedene Technologien, darunter den Microsoft Azure Video Indexer zur Analyse von Video- und Audiodaten sowie Methoden der natürlichen Sprachverarbeitung wie Named Entity Recognition und extraktive Textzusammenfassung. Das Ergebnis der Arbeit ist eine Pipeline zur Extraktion von Fakten, die Personen, Organisationen und Ereignisse erkennt und verknüpfte Informationen anhand eines Schemas, basierend auf dem schema.org-Standard, strukturiert.

Im Zuge dieser Masterarbeit wurde ein Prototyp implementiert, der das Hochladen von Nachrichtenvideos, deren Verarbeitung sowie die Generierung strukturierter Faktenberichte in maschinenlesbarem Format unterstützt. Die Implementierung kombiniert cloudbasierte und lokale NLP-Komponenten, um eine robuste Extraktion der Fakten zu gewährleisten.

Für die Evaluation des Prototyps wurden 31 Nachrichtenvideos des Senders BBC analysiert. Anhand der Ergebnisse konnte festgestellt werden, dass Fakten zu Personen und Ereignissen zuverlässig extrahiert und strukturiert werden. Organisationen wurden vom Prototyp jedoch deutlich schlechter erkannt. Häufige Fehlerquellen konnten bei der Schreibweise von Namen, bei zu unspezifischen oder zu allgemein formulierten Informationen sowie bei unpräzisen Zeitangaben festgestellt werden.

Diese Masterarbeit präsentiert einen Ansatz zur automatisierten Extraktion von Fakten aus Nachrichtenvideos. Die Ergebnisse der Evaluierung des Prototyps verdeutlichen, in welchen Bereichen Verbesserungsbedarf besteht. Diese beziehen sich vor allem auf die Verarbeitung von Organisationen, die korrekte Extraktion von Informationseinheiten sowie von Zeitdaten. In Zukunft könnte der Fokus auf der Implementierung zusätzlicher Entitätstypen, der optimierten Validierung extrahierter Werte sowie der Integration weiterer Sprachen und Nachrichtenquellen liegen.

Contents

Acknowledgements i

Abstract iii

Kurzfassung v

List of Tables xi

List of Figures xiii

Listings xv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Question . . . 1
- 1.3 Outline . . . 2

2 Background 3
- 2.1 From Misinformation to Conflicting Information . . . 3
- 2.2 Fact-Checking . . . 4
- 2.3 News videos . . . 5
- 2.4 Visual Content Analysis . . . 6
  - 2.4.1 Optical Character Recognition . . . 6
  - 2.4.2 Face Recognition . . . 8
- 2.5 Audio Content Analysis . . . 9
  - 2.5.1 Automatic Speech Recognition . . . 9
  - 2.5.2 Speaker Recognition . . . 11
- 2.6 Video Analysis with Microsoft Azure Video Indexer . . . 12
- 2.7 Natural Language Processing . . . 12
  - 2.7.1 Named Entity Recognition . . . 12
  - 2.7.2 Text Summarisation . . . 15
  - 2.7.3 Artificial Intelligence in Natural Language Processing . . . 17
- 2.8 Summary . . . 18

3 Related Work 21
- 3.1 Research on Automated Fact-Checking . . . 21
  - 3.1.1 Misinformation Detection . . . 21
  - 3.1.2 Video-Based Approaches . . . 21

3.1.3 Data Sets . . . 22
3.2 Structured Information Extraction . . . 22
3.2.1 Generating Structured Data with LLMs . . . 23
3.2.2 Risks of Information Processing with LLMs . . . 24
3.2.3 Multimodal Information Extraction . . . 24
3.3 Video Analysis Tools . . . 24
3.4 Summary . . . 25

4 Conceptual Foundation and Design **27**
4.1 Problem Description . . . 27
4.1.1 Context and Motivation . . . 27
4.1.2 Underlying Problem . . . 28
4.1.3 Task Abstraction . . . 28
4.2 Requirements Specification . . . 29
4.2.1 Functional Requirements . . . 29
4.2.2 Non-Functional Requirements . . . 30
4.3 Core Concepts . . . 31
4.3.1 Information Extraction from News Videos . . . 31
4.3.2 Entity-Based Fact Extraction . . . 31
4.3.3 Schema-Based Fact Extraction . . . 32
4.3.4 Fact Report Structure . . . 33
4.4 System Design . . . 34
4.4.1 System Architecture Overview . . . 34
4.4.2 Processing Pipeline . . . 35
4.4.3 Video Analysis Component . . . 36
4.4.4 Content Processing Component . . . 36
4.4.5 Fact Structuring Component . . . 36
4.4.6 Fact Report Generation . . . 37
4.4.7 User Interface . . . 37
4.5 Summary . . . 37

5 Implementation **39**
5.1 Implementation Scope . . . 39
5.1.1 Objective of the Prototype . . . 39
5.1.2 Relation to the Research Questions . . . 39
5.1.3 Limitations . . . 40
5.1.4 Backend and Frontend Responsibilities . . . 40
5.2 System Architecture . . . 40
5.2.1 Overview of Architecture . . . 40
5.2.2 Technology Stack . . . 43
5.3 Backend Implementation . . . 44
5.3.1 Backend Architecture . . . 44
5.3.2 Fact Schemas . . . 46
5.3.3 Data Storage . . . 46

5.3.4 Configuration and Environment Variables . . . 46
5.4 Frontend Implementation . . . 47
5.4.1 Frontend Architecture . . . 47
5.4.2 User Interface Design . . . 48
5.5 Deployment and Execution . . . 55
5.6 Summary . . . 56

6 Evaluation **57**
6.1 Evaluation Data . . . 57
6.1.1 Selection Criteria . . . 57
6.1.2 Data Set . . . 57
6.1.3 Limitations . . . 60
6.2 Automatic Fact Extraction . . . 60
6.2.1 Preparation and Requirements . . . 61
6.2.2 Fact Extraction by the Prototype . . . 61
6.2.3 Fact Report . . . 62
6.3 Manual Fact Validation . . . 63
6.3.1 Evaluation Procedure . . . 64
6.3.2 Evaluation Metrics . . . 64
6.4 Results . . . 64
6.4.1 Entity Detection . . . 64
6.4.2 Detected Properties . . . 67
6.4.3 Qualitative Observations . . . 72
6.5 Discussion . . . 77

7 Conclusion **85**
7.1 Research Question Findings . . . 85
7.2 Limitations . . . 86
7.3 Future Work . . . 86

Bibliography **87**

List of Tables

2.1 spaCy entity labels 14
2.2 Azure entity labels 15

6.1 Data set 60
6.2 False properties: person 69
6.3 False properties: organisation 71
6.4 False properties: event 72
6.5 Fact example: misspelling 73
6.6 Fact example: meaningless "foundingLocation" 74
6.7 Fact example: meaningless "url" 75
6.8 Fact example: unspecific "date" 75
6.9 Fact example: unspecific "endDate" 76
6.10 Fact example: "Friday" as unspecific "date" 76
6.11 Fact example: "recent" as unspecific "date" 77
6.12 Fact example: unspecific "dissolutionDate" 77
6.13 Fact example: unspecific description 77
6.14 Fact example: unspecific "description" value 78
6.15 Fact example: unspecific property value 78
6.16 Fact example: partially incorrect property value for "description" 79
6.17 Fact example: partially incorrect property value for "jobTitle" 79
6.18 Fact example: broadcasting company 79
6.19 Fact example: over-represented broadcasting company 80

List of Figures

4.1 Fact report structure 33
4.2 Fact extraction pipeline 35

5.1 Component diagram 41
5.2 Sequence diagram 42
5.3 User interface 49
5.4 User interface: upload video 50
5.5 User interface: process video 50
5.6 Process video: video selection 51
5.7 Process video: selected video 51
5.8 User interface: analyse content 52
5.9 User interface: show facts 53
5.10 Show facts: fact report 53
5.11 Show facts: facts 54
5.12 Show facts: person facts 54

6.1 Fact report: persons 62
6.2 Fact report: evaluation 63
6.3 Detected entities 65
6.4 Undetected entities 66
6.5 Distinct properties 67
6.6 Property performance: person 81
6.7 Property performance: organisation 82
6.8 Property performance: event 83

Listings

2.1 Example of OCR result 8
2.2 Example of Speech-to-Text result 10

1 Introduction

1.1 Motivation

The increasing availability and speed of news video distribution present both opportunities and challenges for information dissemination [1]. Videos deliver rich, multimodal content by combining speech, visual elements, and textual overlays. However, this information is unstructured and distributed across various channels. As a result, retrieving and verifying key information, such as details about mentioned persons, organisations, or events, is often difficult and time-consuming [2].

This complexity makes it challenging to detect misinformation distributed through videos. Manual fact-checking and analysis of news videos are unsustainable given the growing volume and complexity of digital content, which is increasing the risk and impact of misinformation [3]. While automated fact-checking solutions for text-based news have attracted considerable attention in recent years [4], there remains a gap in addressing the multimodal structure of videos [2]. Efficient, automated extraction and structuring of facts from news videos is essential for improving the reliability and accessibility of information and supporting automated fact-checking applications [5].

This thesis addresses the need for effective, automated methods to extract and organise facts from news videos in order to support trustworthy digital news.

This thesis proposes an approach to automatically extract facts from news videos, converting unstructured audiovisual content into structured, machine-readable fact reports for comparison and verification.

1.2 Research Question

The main research question is: How can information be automatically extracted from news videos? This question is addressed through three parts:

1. Structuring unstructured data: How can the content of news videos be transformed into a technical structure that allows individual information elements to be accessed? This involves identifying and separating information presented through audio and visual elements.

2. Organising key facts: How can extracted information be summarised and organised into concise, non-redundant facts? An efficient organisation ensures that the main aspects of the video are captured without including non-essential details.

3. Prototype implementation for fact extraction: How can a functional system be developed to demonstrate automatic fact extraction from news videos? The prototype focuses on extracting key entities and their associated properties into a structured format.

1.3 Outline

The thesis is structured as follows:

- **Chapter 2 - Background** provides background on misinformation, fact-checking methods, and core technologies in Natural Language Processing and video processing.
- **Chapter 3 - Related Work** reviews related work in automated fact-checking, information extraction, and video analysis.
- **Chapter 4 - Conceptual Foundation and Design** presents the conceptual design of the proposed approach, detailing system requirements, core concepts, and pipeline architecture.
- **Chapter 5 - Implementation** outlines the development process and technical approach used for implementing the prototype.
- **Chapter 6 - Evaluation** presents the main results and discusses the achievements and limitations of the proposed prototype.
- **Chapter 7 - Conclusion** summarises contributions, answers the research questions, and proposes future research directions.

2 Background

2.1 From Misinformation to Conflicting Information

In recent years, the spread of false information in the media and the threats it poses to society have attracted increasing public and scholarly attention [3]. This section outlines various academic approaches to false information, examines its origins, and clarifies key terminology, including *misinformation*, *disinformation*, *fake news*, and *conflicting information*.

Misinformation refers to inaccurate or misleading information, from political conspiracies to misinterpreted pranks. As a general category, misinformation does not require intentional creation or wilful spreading of misleading content [6]. Jiang and Wilson [6] classify misinformation stories into ten types: fabricated and misleading content; satire and parody; legends and tales; conspiracy theories; digital alteration; hoaxes and pranks; clickbait using inflammatory and sensational language; fictional content; bogus scams; and mistakes and errors. In news events from 2010 to 2020, they find that conspiracy theories, fabricated content, and digital manipulation became more common, while legends and tales, hoaxes and pranks, and mistakes and errors declined. These outcomes indicate that misinformation in the news has become more manipulative.

Closely related to this development is the concept of *disinformation*, which is information deliberately created and spread to deceive. Unlike misinformation, disinformation has a clear strategic intent. Disinformation campaigns aim to spread quickly and reach wide audiences, often by targeting novel or highly emotional topics [7].

The term *fake news* attained prominence after the 2016 US presidential election. It is used frequently in public and academic debates. Fake news includes wilfully distributed disinformation, as well as misinterpreted, manipulated, or fabricated content lacking deliberate intent. Whether false information stems from poor journalism or from coordinated disinformation, research shows that it spreads faster and persists longer than corrections. Often, corrections do not reach the original audience, leading to the continued circulation of uncorrected misinformation [8].

Various strategies are used to create misleading content. These include repurposing outdated material, such as presenting old news stories as current events, and digital manipulation that alters images, videos, or audio, distorting the content's original meaning. Another common strategy is to misrepresent authentic content, for example, through video news releases, which are press releases designed to appear as actual video news reports but produced by marketing or political entities. Such practices blur the line between objective reporting and promotional or biased messaging, leading audiences to view crafted content as factual news. Social media and online platforms further increase

these practices, as fast dissemination and viral sharing increase the reach and longevity of misleading information [3].

Within the conceptual framework of this master's thesis, the term *conflicting information* is used instead of misinformation, disinformation, or fake news. This term emphasises that the information differs from other perspectives on the same topic, without judging its accuracy. The thesis does not attempt to verify the truth or origin of information. Instead, it focuses on exposing differences between competing representations. Users determine for themselves what is true or false. Many fact-checking agencies also avoid ruling on validity and do not label information as true or false since most topics are complex and present several points of view on what is right and wrong [9].

2.2 Fact-Checking

This section outlines current knowledge on the detection of conflicting information. Therefore, it provides an overview of different fact-checking methods and summarises their advantages and limitations.

Fact-checking is the process of verifying the accuracy of information presented in written or spoken text [10]. This task is time-critical, as widespread false information can seriously impact society. For example, it can influence votes in elections or affect stock market prices [11]. The spread of misinformation is not new; historically, people have often deliberately distributed false claims. In recent years, information distribution has grown significantly. The emergence of digital media has enabled the quick dissemination of news to expanded audiences. In many cases, false information, whether wilfully distributed disinformation, or misinterpreted, manipulated, or fabricated content as discussed in the previous section, spreads even faster because it is often dramatic or novel [4]. Thus, there is growing interest in fact-checking methods.

Various approaches to fact-checking have emerged in recent years. Zhou and Zafarani [11] describe four main approaches to detecting false information. Knowledge-based methods check if content matches verified facts. Style-based methods examine patterns in news, such as emotional or informal writing. Propagation-based methods study how news spreads online. Source-based methods evaluate news source credibility from creation to distribution.

Although style-based, propagation-based, and source-based methods largely rely on machine learning and deep learning, knowledge-based fact-checking can be performed manually or automatically. Manual fact-checking in knowledge-based methods relies on two primary methods: expert-based fact-checking and crowdsourced fact-checking. Expert-based fact-checking involves domain experts who verify the accuracy of news. Fact-checking websites such as PolitiFact [12] and Snopes [13] employ this approach and provide detailed analyses of news accuracy. This method produces highly reliable results but is less appropriate for large-scale fact-checking because it requires experts in the relevant knowledge domain. In contrast, crowdsourced fact-checking gathers large groups of individuals, often through crowdsourcing platforms such as Amazon Mechanical Turk [14] to assess the accuracy of the content. Although this approach is better suited to

fact-checking a large volume of news stories, it may be less reliable due to likely biases and conflicting verdicts among participants [11]. To summarise, the process of knowledge-based fact-checking requires two steps: fact extraction and knowledge comparison with available information on the topic. Consequently, it is vital to have a reliable source as a reference for fact comparison. Knowledge bases such as DBpedia or Knowledge Vault can serve this purpose [11].

Manual approaches are time-consuming and require human resources. Therefore, efforts are underway to integrate manual fact-checking into automated systems. These systems aim to detect misinformation quickly across large volumes of online content [3]. Automatic fact-checking techniques are primarily based on information retrieval, natural language processing, machine learning, and network theory. Many rely on a standardised representation of facts, using subject-predicate-object triples [11].

Guo et al. [10] describe automated fact-checking as a process consisting of three main steps: claim detection, evidence retrieval, and claim verification. The first stage, claim detection, selects claims worthy of fact-checking. This requires a decision-making process to determine whether the claim is evidence-based and verifiable, as personal views or statements lacking available evidence cannot be verified. Additionally, the temporal relevance of claims must be considered, as outdated claims are generally less important to fact-check than those concerning current events. In the second stage, it is necessary to retrieve evidence about the claim from an alternative source. This can be information about the claim stored in a knowledge base or complementary information retrieved from images, tables, or metadata. The final stage, claim verification, involves labelling the validity of the claim. While a simple classification as *true/false* is possible, the complexity of real-life topics often requires more detailed labels, such as *supported/refuted by evidence* or *conflicting/non-conflicting*. To develop trust in automated systems, it is essential not only to present the verdict but also to provide a transparent justification of the decision, including the evidence that led to the final conclusion.

A frequent challenge in the automated fact-checking process is incomplete knowledge of certain topics. This particularly concerns facts about recently published news events, as reliable reference information is often not yet available [11].

2.3 News videos

Over the last decade, there has been a significant change in the consumption and production of news videos. Traditionally, broadcasters created and published their news videos exclusively for television. Consumers watched these videos on television at scheduled broadcast times, mainly at home. As a result, news videos were designed for a largely uniform consumption environment [15]. The format of news videos has always varied by channel and by the cultural and linguistic choices of each country. Factors such as the role of the anchor, the emphasis on specific topics, and the overall structure of the news program differed accordingly [16].

However, in the mid-2000s, digital technologies enabling online video streaming began to affect the production and consumption of news videos. Since then, improved internet

connections and new streaming devices have led to a shift toward online presentation formats [17]. Although broadcasters had previously prioritised television-based formats, they have increasingly adapted their content to mobile-friendly formats [18].

The number of consumers who prefer to watch news videos online has been steadily increasing for almost ten years, as demonstrated by Kalogeropoulos in a 2016 study. He examined five European countries and the United States, comparing user preferences for news video consumption, and found that online news videos are gaining in importance over television news videos. He further noted that these videos are accessed either through broadcasters' websites or via social media platforms, with the latter becoming increasingly important [18].

The growing public interest in online news videos has raised awareness among broadcasting stations of the need to adapt to the digital environment [17]. In response to these changes, broadcasters are increasingly tailoring their content to digital platforms. Traditional broadcasting organisations now produce mobile-optimised formats, often featuring shorter clips designed for distribution on social media in order to reach broader audiences [1].

As time has become increasingly critical in digital news consumption, broadcasting stations have also adopted interactive features such as push notifications and algorithmic scheduling in mobile applications, helping maintain audience engagement in a highly competitive digital news environment. Research indicates that the lifecycle of news content is often brief, with most user engagement occurring within the first few hours after publication. This rapid consumption cycle highlights the importance of fast news delivery and the competitive pressure on broadcasters to provide timely and relevant content [1].

Finally, trust in news videos remains a complex and evolving issue. Users' perceptions of credibility are shaped by factors such as the source of the content, the transparency of editorial processes, and individual engagement with news. Although established news brands often retain greater trust than social media sources, peer-to-peer sharing and algorithms increasingly influence audience judgments, particularly among younger viewers. Long-term studies show that as people increasingly rely on social media for news, trust in traditional news institutions tends to decline, highlighting the need for broadcasters to adapt their practices to maintain credibility [19].

2.4 Visual Content Analysis

This section outlines essential techniques for analysing visual content and provides background for the considerations regarding the prototype implemented for this master's thesis.

2.4.1 Optical Character Recognition

Optical Character Recognition (OCR) is a technology that converts text into a machine-readable format. It can process scanned paper documents or text-containing images,

making the content searchable and editable. Developed in the 1950s, OCR was initially designed to digitise books and documents [20]. Today, it plays a crucial role across a wide range of industries and applications [21].

A primary objective of OCR is to convert printed text in standard fonts, such as those used in books. However, processing printed text causes notable challenges when documents feature unusual layouts, poor formatting, or complex backgrounds, which can compromise recognition accuracy. Another major challenge is recognising handwritten text and converting it into machine-readable characters. This task is particularly demanding due to the high variability in handwriting styles, requiring advanced machine learning models trained on diverse handwritten samples [21]

Furthermore, OCR is increasingly applied to images captured in real-world environments, such as street signs and product labels. In these contexts, OCR systems must employ computer vision algorithms able to handle dynamic settings, varying lighting conditions, and image distortions [21].

In an initial step, OCR algorithms preprocess the image to enhance text readability. This typically involves noise reduction, contrast enhancement, and image alignment. Subsequently, the algorithm detects text regions by identifying characteristic structures and patterns corresponding to words or characters. These regions are then analysed in detail, and the recognised characters are combined to form words. In a final step, the extracted text is processed in context and validated using grammatical rules and spell-checking techniques to reduce recognition errors [21].

In this master's thesis, OCR is employed to extract textual information from news videos. This includes news headlines, interview partners' names, and broadcasting station logos. For this purpose, the Microsoft Azure OCR engine is used [22].

Optical Character Recognition with Microsoft Azure

For the prototype implementation, the OCR functionality provided by *Microsoft Azure AI Vision* is used as an integrated component of *Microsoft Azure Video Indexer*. The service is part of *Azure AI Services* and provides a cloud-based Optical Character Recognition engine. It enables the extraction of printed and handwritten text from images and documents using deep learning-based models [22].

Azure distinguishes between OCR optimised for general images and OCR optimised for document processing. The *OCR for Images* engine (Vision API v4.0) is designed for images such as photographs of signs, posters, and video frames. It provides a synchronous REST interface suitable for near-real-time applications. In contrast, the *Read OCR* functionality is optimised for scanned documents and PDF files and is typically accessed via an asynchronous workflow [23]. In this master's thesis, the image-based OCR interface is applied since it is part of the integrated Azure Video Indexer [22].

The Azure OCR engine supports multiple languages and can detect both printed and handwritten text. The service returns structured JSON output [23].

Listing 2.1 presents an example of an OCR result generated by the OCR component integrated into Azure Video Indexer. For each detected word or line, the service returns the recognised text string, a confidence score indicating the reliability of the prediction,

the bounding box coordinates specifying the location within the image, the detected source language, and the corresponding time interval during which the text was visible in the video.

```
{
    "id": 13,
    "text": "The Rise and Fall of Michelle Mone / BBC",
    "confidence": 0.9959,
    "left": 679,
    "top": 556,
    "width": 413,
    "height": 25,
    "angle": 0,
    "language": "en-US",
    "instances": [
        {
            "adjustedStart": "0:00:30.24",
            "adjustedEnd": "0:00:35.08",
            "start": "0:00:30.24",
            "end": "0:00:35.08"
        }
    ]
}
```

Listing 2.1: Example of OCR result

2.4.2 Face Recognition

Face recognition is a biometric authentication technology that identifies or verifies individuals based on their facial features. Biometric characteristics are unique to each individual and difficult to replicate, thus providing a secure method of identification. Although earlier biometric modalities, such as fingerprint and iris recognition, have been widely adopted, facial recognition has emerged as a prominent component of modern biometric authentication systems. It is applied in various domains, including access control, security, surveillance, and payment systems [24].

Although humans effortlessly recognise faces, this task constitutes significant challenges for computer systems. Automated face recognition requires analysing complex visual patterns that vary significantly due to factors such as illumination conditions, viewing angles, and changes in facial expressions. These variations make reliable and accurate recognition a challenge [24].

Face recognition can be applied to both images and videos. Image-based systems primarily identify people based on static facial features. In contrast, video-based systems consider not only static characteristics but also temporal information, such as changes in appearance over time and dynamic facial movements [25].

The first step in a face recognition pipeline is detecting human faces in an image or video frame. Once a face has been detected, face recognition systems typically operate in two distinct phases. The first phase, face verification, determines whether a given face corresponds to a specific claimed identity by comparing the extracted features of the input

image with those of a predefined individual. The second phase, face identification, aims to determine the identity of an unknown individual by comparing their facial features against a database of known identities. In this process, the system analyses various facial attributes, such as the shapes of the eyes, nose, and mouth, as well as the overall facial geometry, to identify the closest match within the database [25].

In the context of this master's thesis, the Microsoft Azure AI Face Service was initially intended to be used to identify individuals in news videos [26]. However, during the conceptual phase of this work, the service policies regarding the use of face recognition were revised. Due to increased restrictions motivated by privacy concerns, Azure limited the use of face recognition [27]. Consequently, face recognition was not included in the prototype.

2.5 Audio Content Analysis

This section presents key technologies for analysing audio content and provides the background for the considerations regarding the implemented prototype of this master's thesis.

2.5.1 Automatic Speech Recognition

Automatic Speech Recognition (ASR), also referred to as Speech-to-Text (STT), is a technology that transcribes spoken language into written text. It has become important in various domains, including healthcare, customer service, education, and entertainment. ASR facilitates more efficient communication between humans and machines and, in certain contexts, among humans themselves [28].

A primary objective of human–machine communication is to enhance the user experience across different applications. Typical use cases include voice search, which enables users to retrieve information in natural language; virtual assistants integrated into devices such as smartphones; and voice-controlled home automation systems. In the context of human–human communication, ASR supports applications such as voice message dictation and real-time translation between different languages [28].

Research on ASR dates back to the 1950s. Early systems relied on pipeline-based architectures that processed speech using separate acoustic models, language models, and predefined phoneme inventories. These components operated independently, each performing a distinct function within the recognition process. Since then, ASR technology has evolved considerably and has benefited significantly from advances in artificial intelligence. Modern ASR systems increasingly employ end-to-end architectures that integrate multiple processing steps into a single model, thereby optimising speech recognition. The adoption of machine learning techniques, particularly deep neural networks, has led to substantial improvements in the accuracy, robustness, and efficiency of ASR systems [29].

Nevertheless, several challenges remain. Real-time speech processing is demanding, as it requires precise temporal segmentation without fragmenting individual words [29]. Additionally, ASR systems frequently struggle in noisy environments, where background

sounds interfere with recognition, and in scenarios involving overlapping speech, in which multiple speakers contribute simultaneously to the audio stream [30].

In this master's thesis, the Speech-to-Text functionality of the Microsoft Azure Speech Service is used to generate transcripts of news videos [31].

Automatic Speech Recognition with Microsoft Azure

For the prototype implementation, the Speech-to-Text functionality provided by *Microsoft Azure AI Speech* is used as an integrated component of *Microsoft Azure Video Indexer*. The service is part of *Azure AI Services* and provides a cloud-based Automatic Speech Recognition engine based on deep neural network models. It enables the transcription of spoken language into text and supports both real-time and batch processing scenarios [31].

The Azure Speech-to-Text service provides multiple recognition modes. *Real-time transcription* is designed for low-latency applications such as live captioning or voice assistants. Audio data is streamed to the service, and recognition results are returned continuously. In contrast, *batch transcription* is optimised for asynchronous processing of large audio or video files. In this mode, audio files stored in cloud storage are processed, and the results are made available upon completion of the recognition task [31]. In this thesis, the batch transcription mode of the Azure Speech-to-Text service is used within Azure Video Indexer to generate transcripts for the analysed videos [27].

The service offers automatic language detection and multilingual speech recognition. In addition, Azure provides advanced features such as speaker diarization, which distinguishes between different speakers within a single audio stream, and word-level timestamps that specify the temporal position of each recognised token. Furthermore, recognition accuracy can be improved through *Custom Speech* models that allow domain-specific adaptation using user-provided training data. The Speech-to-Text API can be accessed via a REST interface, and returns structured JSON output [31].

Listing 2.2 presents an example of a transcription result generated by the Azure Speech-to-Text service integrated into Azure Video Indexer. For each recognised segment, the service provides the transcribed text, a confidence score, the identified speaker, the detected source language, and the corresponding time interval.

```
1
2 {
3     "id": 1,
4     "text": "Now the High Court has ruled that a company linked to the
      former Conservative peer Baroness Mona and her husband Doug Barrowman
      must repay the UK government almost \u00a3122 million for breaching a
      contract to supply surgical gowns during the pandemic.",
5     "confidence": 0.8449,
6     "speakerId": 1,
7     "language": "en-US",
8     "instances": [
9         {
10            "adjustedStart": "0:00:00.72",
11            "adjustedEnd": "0:00:17",
```

```
            "start": "0:00:00.72",
            "end": "0:00:17"
        }
    ]
}
```

Listing 2.2: Example of Speech-to-Text result

2.5.2 Speaker Recognition

Speaker Recognition refers to the task of associating a speech signal with its corresponding speaker. The human voice is influenced by individual physical and behavioural characteristics, which create a unique combination of features such as frequency, pitch, and speech energy. Consequently, a voice carries speaker-specific information that can be used for automatic identification. Research in speaker recognition dates back to the 1960s and builds upon advances in Automatic Speech Recognition (ASR), as outlined in the above section. Early studies focused primarily on text-dependent scenarios, in which speakers read predefined texts. Over time, methodological advancements enabled the recognition of speakers from spontaneous speech, consequently broadening the range of potential applications [32].

The field of Speaker Recognition is commonly divided into three subareas: Speaker Verification, Speaker Identification, and Speaker Diarization. Speaker Verification aims to determine whether a given speech sample matches a claimed identity stored in a preexisting database. It typically assumes that the analysed audio segment contains speech from a single speaker and requires prior enrollment of that speaker in the system. In contrast, Speaker Identification aims to determine the identity of an unknown speaker by comparing a speech sample to a predefined set of registered speakers. While Speaker Verification evaluates the validity of a claimed identity, Speaker Identification selects the most likely match from a group of known speakers. Speaker Diarization, on the other hand, addresses audio recordings containing speech from multiple speakers. Its primary objective is to segment the audio stream and attribute each segment to the corresponding speaker. A major challenge in Speaker Diarization is handling overlapping speech segments, where multiple speakers speak simultaneously. Unlike Speaker Verification and Speaker Identification, Speaker Diarization does not require a prior enrollment process and can be applied to previously unseen speakers. It is also suitable for near real-time speech processing scenarios [33].

The Microsoft Azure Speaker Diarization service [34] was initially considered for the prototype implementation to obtain insights into the speakers in news videos. However, during the conceptual phase of this master's thesis, an entity-based approach was selected, as described in Chapter 4. Instead of attributing extracted key phrases to specific speakers, information about entities was collected throughout the video and persisted as structured fact representations. This approach was chosen over a speaker-based attribution strategy.

2.6 Video Analysis with Microsoft Azure Video Indexer

Microsoft Azure Video Indexer [35] is a cloud-based video analytics service that uses artificial intelligence to extract structured insights from audio and video content. The service is part of the Azure AI platform and enables automated analysis of unstructured media using built-in AI models [27].

Azure Video Indexer integrates multiple analysis pipelines that combine audiovisual and textual data into a collection of insights. Its capabilities include, for example, speech transcription, object and scene detection, key-frame extraction, and optical character recognition. The service can be accessed via a web portal, a REST API, or via SDKs [27].

Each processed video is annotated with structured data, including transcripts with time codes, detected visual text, scene segmentation markers, and confidence scores for each identified insight. The result is available either as a graphical representation in the web portal or as a structured JSON file. These outputs can be integrated into further applications such as indexing, content retrieval, or semantic search [27].

In this master's thesis, *Microsoft Azure Video Indexer* is utilised in the prototype implementation to analyse raw video material and obtain structured information about audiovisual content and related metadata. The extracted data is subsequently transformed into a uniform fact structure.

2.7 Natural Language Processing

This section introduces key technologies in Natural Language Processing (NLP) that are utilised in the implementation of this master's thesis.

2.7.1 Named Entity Recognition

Named Entity Recognition (NER), also known as Entity Extraction, is a core task in Natural Language Processing (NLP) that identifies and classifies words or phrases into predefined categories, such as persons, locations, temporal expressions, and organisations. These categories are commonly referred to as entities. For example, in the sentence *David was born in New York,* a NER system would classify *David* as a person and *New York* as a location [36].

NER is a fundamental technique for structuring unstructured text by transforming it into machine-readable, structured information. Its applications cover a wide range of domains, including information retrieval, machine translation, question-answering systems, and text summarisation [37].

NER systems can be implemented using different methodological approaches. A traditional approach is rule-based, in which linguists or domain experts define rule templates based on linguistic features such as part-of-speech tags, syntactic patterns, punctuation, and grammatical structures. These rules are typically domain-specific, enabling high precision within a narrowly defined context. However, these approaches are difficult to transfer to new domains and require considerable manual effort for adaptation [37].

In recent years, deep learning-based approaches have become predominant. These methods automatically learn relevant features from large datasets, reducing the need for manual feature engineering. Neural architectures, such as recurrent neural networks and transformer-based models, have significantly improved the performance and robustness of NER systems by capturing contextual dependencies within text [36].

Despite these advances, several challenges remain. NER systems require large annotated datasets for training, which are time-consuming and costly to produce, particularly when domain expertise is necessary. Additionally, ambiguity in language, such as words with multiple meanings, and informal language, as used in social media content, increases the complexity of entity recognition. The handling of noisy or misspelt text, for example, due to inaccuracies or human input errors, further complicates the accurate identification and classification of named entities [37].

In this master's thesis, Named Entity Recognition is performed using both the spaCy library [38] and Microsoft Azure Language Services [39] to detect person, organisation, and event entities mentioned in videos.

Named Entity Recognition with spaCy

For the prototype implementation, Named Entity Recognition is performed using the *spaCy* library, an open-source Natural Language Processing toolkit for Python. SpaCy provides the *EntityRecognizer*, an NER component that identifies and classifies text spans corresponding to named entities such as persons, organisations, locations, dates, and other predefined categories [38]. The full list of spaCy labels available for the version used in this master's thesis is shown in Table 2.1.

In spaCy, the NER component is implemented as a trainable pipeline module that can be integrated into an NLP processing workflow. The input text is first tokenised and represented as a document object. The NER component then processes this document and predicts non-overlapping labelled spans, which are stored in the entities attribute. Each identified entity corresponds to a sequence of tokens that represent a concept or object in the real world, such as the name of a person or the title of an organisation [38]. Each entity is associated with its text span, label, and position within the document. This structured output can be used for tasks such as semantic analysis, indexing, or information extraction [38].

The underlying model relies on statistical machine learning methods and learned word representations to assign entity labels. This enables the recognition of entities without the need for manually defined rules. Additionally, spaCy supports customisation. Users can extend the set of recognised entity types and retrain models using domain-specific annotated data when required [38].

Named Entity Recognition with Microsoft Azure

For the prototype implementation, the Named Entity Recognition functionality provided by *Microsoft Azure AI Language* is used as an additional approach to spaCy. The service is part of *Azure AI Services* and offers a cloud-based Natural Language Processing engine

Category	Description
Cardinal	Numerals that do not fall under another type
Date	Absolute or relative dates or periods
Event	Named hurricanes, battles, wars, sports events, etc.
Fac	Buildings, airports, highways, bridges, etc.
GPE	Countries, cities, states
Language	Any named language
Law	Named legal documents
Loc	Non-GPE locations, such as mountain ranges or bodies of water
Money	Monetary values, including units
Norp	Nationalities, religious groups, or political groups
Ordinal	Ordinal numbers (e.g., "first", "second")
Org	Companies, agencies, institutions, etc.
Percent	Percentages, including "%"
Person	People, including fictional characters
Product	Objects, vehicles, foods, etc. (excluding services)
Quantity	Measurements, such as weight or distance
Time	Time expressions smaller than a day
Work of art	Titles of books, songs, and similar works

Table 2.1: spaCy entity labels

for detecting and classifying named entities in unstructured text. The NER component is based on deep learning models trained on large multilingual datasets [39].

The Azure NER service identifies predefined entity categories in text. These include common entity types such as persons, organisations, locations, dates, times, quantities, and events. The full list of available Azure labels for the version used in this master's thesis is shown in Table 2.2. In contrast to spaCy, described in the previous section, Azure additionally provides subcategories that further refine the main entity categories [39].

In addition to general-purpose entity recognition, Azure provides specialised recognition modes. For example, *PII entity recognition* is designed to detect personal information such as email addresses, phone numbers, or identification numbers. Furthermore, domain-specific models are available for extracting healthcare-related entities, enabling the identification of medical terms, symptoms, and treatments [39].

The service supports multiple languages and can be accessed via REST interfaces or SDKs for various programming languages. Each detected entity is returned together with its textual representation, category, optional subcategory, confidence score, character offset, and length within the original document [39].

Category	**Subcategory**
Address	—
DateTime	Date
DateTime	DateRange
DateTime	Duration
DateTime	Set
DateTime	Time
DateTime	TimeRange
Email	—
Event	—
IP	—
Location	GPE
Location	GeographicFeature
Location	Structural
Organization	—
Person	—
PersonType	JobTitle
PhoneNumber	—
Product	—
Quantity	Age
Quantity	Currency
Quantity	Dimension
Quantity	Number
Quantity	Ordinal
Quantity	Percentage
Quantity	Temperature
Skill	—
URL	—

Table 2.2: Azure entity labels

2.7.2 Text Summarisation

Text summarisation is a Natural Language Processing task that produces a shorter version of a text, though preserving its essential information. Because manual summarisation is time-consuming, automatic summarisation is a suitable tool to improve efficiency. Text summarisation is useful in many NLP applications, including question answering, news summarisation, and headline generation. In addition, automatically generated summaries can be used as a preprocessing step to shorten long documents, thereby reducing resources for follow-up tasks [40].

Research on automatic text summarisation began in the 1950s, with statistical methods that identified important sentences based on word frequency and positional features. These early systems relied on simple heuristics and analysed surface-level features. With

advances in machine learning and deep learning, summarisation methods have evolved to neural architectures that capture semantic relationships and contextual information in documents [41].

Automatic text summarisation can be distinguished into extractive and abstractive summarisation [40].

Extractive summarisation selects the most relevant sentences or text segments directly from the original document without altering the wording. Traditional extractive methods use statistical features such as sentence position or graph-based ranking algorithms to estimate sentence importance. Although extractive methods are easier to implement, they cannot capture context, which can reduce the coherence of the summary [40].

Abstractive summarisation, in contrast, generates new sentences that condense the original text. Rather than copying text fragments, abstractive summarisation aims to capture a document's meaning and produce a coherent summary [40]. More recently, large language models (LLMs) have been applied to capture relationships across entire documents and generate fluent summaries. These models often produce more concise and readable summaries than extractive systems. However, they require more resources and larger training datasets and are more prone to hallucinations, in which the generated information is not present in the source text [41].

Despite considerable progress, text summarisation remains a challenging task. A key challenge is selecting effective content, as the system must identify the most relevant information while avoiding redundancy. Keeping coherence and logical consistency, particularly in long documents, is another major difficulty. For abstractive models, ensuring factual faithfulness to the source text remains a key issue. Furthermore, evaluating summary quality is complex. Automatic metrics do not always correlate well with human judgments regarding readability, informativeness, and factual correctness [41].

In this master's thesis, text summarisation is performed with the *spaCy* [42] library and *Azure Language Services* [43]. For both tools, an extractive summarisation method is applied.

Text Summarisation with spaCy

For the prototype, the *spaCy* [42] library is used for text summarisation. Although spaCy does not provide a built-in text summarisation functionality, it offers the necessary building blocks to implement it.

In the prototype, an extractive summarisation method is implemented using spaCy's NLP components, including tokenisation, part-of-speech tagging, and sentence segmentation.

First, the input text is processed by splitting the document into tokens and sentences. Stop words and punctuation are removed. Only tokens with specific part-of-speech tags, including nouns, adjectives, and verbs, are considered as candidate keywords.

Next, a frequency-based scoring mechanism is applied. The occurrences of the selected keywords are counted and normalised by the maximum word frequency in the document. This normalisation prevents sentence scores from being disproportionately influenced by frequently occurring terms. Each sentence is then assigned a score based on the sum of the

normalised frequencies of the keywords it contains. Sentences with a higher concentration of frequent and semantically relevant terms receive higher scores.

Finally, the highest-ranking sentences are extracted and concatenated to produce the final extractive summary [42].

Text Summarisation with Microsoft Azure

For the prototype implementation, the text summarisation functionality provided by *Microsoft Azure AI Language* is used as an additional approach to spaCy. The service is part of *Azure AI Services* and can be accessed through REST endpoints or language-specific SDKs [43].

Azure Text Summarisation supports both abstractive and extractive summarisation. Extractive summarisation is available as a feature of Microsoft Azure AI Language, while abstractive summaries can be generated using Azure's generative models, for example, through Azure OpenAI Service [43].

In the prototype, extractive summarisation is implemented using the Microsoft Azure AI Language Python SDK. The service analyses the submitted document and returns a ranked list of sentences that best represent the text's core content. These sentences are extracted from the original document and ordered by relevance. The number of sentences included in the summary is dynamically calculated based on the total number of sentences in the input text, ensuring that the summary length scales proportionally with the document size [43].

2.7.3 Artificial Intelligence in Natural Language Processing

Artificial Intelligence (AI) refers to computational systems that are capable of performing tasks that typically require human intelligence, such as reasoning, learning, perception, and language understanding. Within AI, Natural Language Processing focuses on enabling machines to process, interpret, and generate human language in both written and spoken form. In recent years, advances in machine learning, particularly deep learning, have significantly transformed the field of NLP [44].

Traditional NLP systems relied primarily on rule-based methods and manually engineered features. Linguistic rules, dictionaries, and syntactic patterns were explicitly defined to analyse text. Although such approaches achieved high precision in narrowly defined domains, they were difficult to scale and adapt to new contexts. With the increasing availability of large-scale text corpora and computational resources, data-driven approaches have gradually replaced rule-based systems [44].

A major milestone in NLP was the introduction of neural network architectures, especially transformer-based models. Prominent examples include GPT, developed by OpenAI, and BERT, developed by Google. These models significantly improved the performance of NLP applications by learning contextual language representations. As a result, they can be adapted to a wide range of tasks without requiring task-specific rule design [45].

The progress in deep learning has enabled the development of advanced applications such as chatbots and virtual assistants, which are now widely used across domains, including finance, healthcare, and transportation. Modern systems can process complex queries and generate contextually appropriate responses, thereby improving human–machine interaction [44].

Despite their versatility, AI models also present limitations. Their outputs are influenced by the data on which they were trained and may therefore reflect biases present in the training corpora. A major challenge is the over-representation of English data compared to other languages, as well as likely biases related to social aspects such as gender, race, or socioeconomic background [45].

In this master's thesis, the *Microsoft Azure OpenAI Service* [46] is used to transform previously extracted information from the video content into a uniform fact representation.

OpenAI with Microsoft Azure

For the prototype implementation, the *Microsoft Azure OpenAI Service* [46] is used to provide AI-based Natural Language Processing. The service provides access to large language models developed by *OpenAI* on Microsoft Azure's cloud infrastructure.

The Azure OpenAI Service provides REST-based API endpoints that allow developers to submit prompts and receive model-generated responses in structured JSON format. Model deployments are created within an Azure resource, and each deployment is associated with a specific model version and configuration [46].

In this master's thesis, GPT-4.1 is deployed within Microsoft Azure and accessed through the Chat Completions API using the Python SDK. Each request specifies the Azure endpoint, the model deployed, and the API version. The conversation's context is defined by structured messages with roles such as *system*, *user*, and *assistant*. The *system* message defines the model's behaviour and constraints, for example, instructing it to extract structured facts. The *user* message contains the actual input text, such as the transcript of a news video. To ensure structured output, predefined JSON schemas can be specified for each request, requiring the model to return responses in a valid JSON format suitable for further automated processing. In addition, configurable parameters such as temperature and token limits influence the randomness and length of the generated output [46].

2.8 Summary

This chapter provided the scientific background for this master's thesis. It first clarified the central terminology regarding false and misleading information. Then, it outlined existing fact-checking approaches, distinguished between manual and automated methods, and highlighted key challenges, including scalability, time constraints, and incomplete knowledge bases.

In addition, the chapter examined the transformation of news video production and consumption over recent years, emphasising the growing importance of online platforms,

rapid dissemination, and factors influencing trust in news videos.

Furthermore, relevant technologies for visual and audio content analysis were introduced, including Optical Character Recognition, Face Recognition, Automatic Speech Recognition, and Speaker Recognition.

Finally, the chapter presented core Natural Language Processing methods, including Named Entity Recognition, Text Summarisation, and Artificial Intelligence.

3 Related Work

3.1 Research on Automated Fact-Checking

This section provides an overview of current research on automated fact-checking. After introducing general approaches, it narrows the focus to video-based and multimodal methods, and then presents widely used data sets for misinformation detection.

3.1.1 Misinformation Detection

In the digital age, the spread of fake news has become a major challenge. While deliberate distribution of false information is not a new phenomenon, digital platforms have significantly increased its speed and reach [8].

False information spreads quickly, whether it is deliberate disinformation or unintended content that is misinterpreted, manipulated, or fabricated. Its often sensational character facilitates this spread [8]. Social media platforms amplify the issue by presenting professional journalism, personal opinion, and political advertising in similar formats. As a result, users may struggle to distinguish credible from unreliable sources [9].

To address this challenge, researchers have developed automated fact-checking systems that often rely on Natural Language Processing to analyse textual patterns and assess the credibility of claims. As a result, text-based methods now represent one of the most advanced and well-studied areas of automated fact-checking [4].

However, misinformation is increasingly distributed through images and videos. As a result, recent research has shifted to multimodal systems that combine text, audio, visual information, and metadata [2]. Since multimodal content is often perceived as more credible than simple text, this increases the need for systems capable of analysing multiple modalities simultaneously [5].

3.1.2 Video-Based Approaches

Several studies focus specifically on fact-checking in video contexts.

In their research, Choi and Ko [47] emphasise the importance of domain knowledge for detecting fake news videos. Their approach integrates video metadata, especially user comments, as additional evidence. Comments are considered valuable because they may contain contextual information or contradicting statements. To address rapidly changing topics, the authors propose a feature selection method and an automatic domain knowledge construction process, enabling the system to adapt to new subject areas.

Peng Qi et al. [48] argue that news videos are published during ongoing events. Their approach compares factual statements across multiple videos related to the same event.

The authors claim that since major events often lead to the production of many similar news videos, linking and comparing these sources can help identify inconsistencies or conflicting claims.

Akhtar, Schlichtkrull et al.[5] propose a framework for multimodal fact-checking that combines text, images, video, and audio. They argue that multimodal disinformation is especially powerful because it appears more trustworthy, spreads quickly, and can be easily generated using modern AI tools. The framework comprises three main phases: claim detection, evidence retrieval, and verdict generation with justification. However, the authors identify several current limitations, including difficulties in extracting claims from complex media, retrieving non-textual evidence, handling multiple languages, and detecting manipulated content.

Another example is presented by Rayar [2], who proposes a multimodal system for analysing political videos. Specifically, the approach combines Face Recognition, Speech Recognition, and NLP techniques to identify and verify claims made in video content.

3.1.3 Data Sets

The development and evaluation of automated fact-checking systems strongly depend on extensive data sets. To support robust research, several benchmark data sets are widely used:

- **FEVER**: The data set contains more than 185,000 claims extracted from Wikipedia. It is an important benchmark for training and evaluating text-based fact-checking systems. The classification uses the labels *supported*, *refuted*, or *not enough info* [49].
- **FEVEROUS**: The data set is an extension of FEVER that includes structured information such as tables and lists in addition to plain text. It consists of approximately 87,000 verified claims. The classification labels are the same as those used in the FEVER data set [50].
- **LIAR**: The data set consists of approximately 13,000 politically oriented claims. The truth labels are fine-grained and also annotate partly true or false claims. It is designed to detect misinformation in text [51].

3.2 Structured Information Extraction

Information Extraction (IE) is a main area of Natural Language Processing. Its goal is to extract relevant information from unstructured text and turn it into a structured, machine-readable format. As unstructured data, such as news articles, scientific papers, and social media posts, grows, automated structuring has become essential, since manual processing is not scalable [52].

Typical IE tasks include Named Entity Recognition (NER), Named Entity Linking (NEL), Coreference Resolution (CR), Relation Extraction (RE), and Knowledge Base

Reasoning. These tasks form the foundation for applications such as question answering, automatic summarisation, and digital assistants [52].

This master's thesis focuses on extracting information from video content. The main challenge is to identify relevant information and transform it into a structured fact representation suitable for further processing, such as fact comparison. As outlined in Chapter 4, analysis begins with the raw video material. Next, Named Entity Recognition is applied to the audio transcript and on-screen text obtained via OCR, defining persons, organisations, and events as key entity types. Finally, Artificial Intelligence based on Large Language Models (LLMs) organises the extracted information about these entities into a unified fact structure.

This section reviews related work in Information Extraction. First, it presents methods for extracting structured data from unstructured sources. Then, it discusses risks of using LLMs and highlights the benefits of integrating multimodal data in the extraction process.

3.2.1 Generating Structured Data with LLMs

The development of Information Extraction (IE) methods can be divided into three main phases: rule-based systems, statistical machine learning, and deep learning. Rule-based approaches achieve high precision in well-defined domains but require extensive manual effort, making them hard to adapt to new tasks. Statistical methods, such as Hidden Markov Models (HMM), Support Vector Machines (SVM), and Conditional Random Fields (CRF), reduce the need for manually written rules, though they still depend on large annotated data sets. Deep learning approaches automatically learn relevant features from data. These methods require high computational resources and large training data sets. However, they significantly improve accuracy and generalisation. Architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), Graph Neural Networks (GNNs), and Transformer-based models enable end-to-end solutions for entity and relation extraction [53].

Large Language Models offer new possibilities for structured information extraction. Many applications, such as agent-based systems and IE pipelines, require structured outputs, such as JSON, for reliable processing [54].

However, a key challenge remains: LLMs often generate outputs that do not strictly adhere to predefined schemas. This reduces the reliability of automated systems. Existing approaches either rely on prompt engineering, which can be unstable, or on constrained decoding, which guarantees structural correctness but can increase latency and limit flexibility [54].

To address the problem of unreliable structured outputs, Wang, Shen et al. [54] propose a post-processing method to convert LLM-generated unstructured outputs into structured formats, such as JSON. Their results show improved compliance with predefined schemas and preservation of content meaning. The authors argue that post-processing can make LLM-generated structured outputs more reliable and suitable for information extraction systems [54].

3.2.2 Risks of Information Processing with LLMs

DeVerna, Yan et al. [55] show that although LLMs are technically powerful, their use in sensitive information processes can lead to unintended effects. In their study, an LLM correctly identified many false political headlines, but was less reliable when evaluating true ones.

More importantly, the use of AI-generated fact-checks did not significantly improve participants' ability to distinguish between true and false information. In some cases, AI feedback even strengthened false beliefs or reduced trust in accurate news. In addition, participants who could voluntarily use AI fact-checks were more likely to share content, regardless of whether it was true or false [55].

The authors argue that high technical accuracy does not automatically lead to positive social impact. For structured Information Extraction systems, this means that not only formal correctness and semantic quality should be considered, but also potential user-related effects [55].

3.2.3 Multimodal Information Extraction

In their 2022 study, Yang, Wu et al. [53] analyse recent developments in Information Extraction research, with a growing focus on multimodal approaches. These approaches combine different data types, such as text, images, audio, and video, to improve information extraction.

The authors show that integrating visual and linguistic information can improve the accuracy of entity and relation extraction. However, challenges remain, particularly in effectively combining different modalities, handling irrelevant or noisy information, and ensuring good generalisation across domains [53].

According to the study [53], future research directions include developing lightweight and efficient models, integrating external knowledge sources, and designing multi-task and multimodal systems. The overall goal is to build robust, scalable IE systems that can extract structured information from complex, diverse data sources.

3.3 Video Analysis Tools

As outlined in Chapter 2, this master's thesis uses *Microsoft Azure Video Indexer* [35] as the central preprocessing component for raw video material. The service transforms audiovisual content into a structured, machine-readable format that can subsequently be processed by NLP methods. It automatically generates transcripts using speech-to-text, detects on-screen text via OCR, and extracts structured metadata such as keywords and timestamps. All extracted information is provided in JSON format, which facilitates integration into the fact extraction pipeline.

An alternative solution is the *Google Cloud Video Intelligence API* [56]. The service applies machine learning models to perform object detection, shot boundary detection, label annotation, speech transcription, and text detection within video streams. Similar to Azure Video Indexer, it generates annotations based on timestamps. However, certain

features, such as speaker diarization and face recognition, require integrating additional Google Cloud services.

Another comparable platform is *Amazon Rekognition Video* [57], which is part of *Amazon Web Services* (AWS). It supports object and scene detection, face recognition, celebrity recognition, activity detection, and content moderation. For speech transcription, it must be combined with an additional AWS service.

Clarifai [58] provides customizable AI models for image and video recognition, allowing developers to train domain-specific classifiers. Although it offers high flexibility, it does not provide an integrated, out-of-the-box solution comparable to Azure Video Indexer. Achieving similar functionality would require combining multiple models and implementing additional processing logic.

For speech transcription, the Speech-to-Text API of *Rev.ai* [59] offers automated transcription services. However, it focuses primarily on speech recognition and does not include integrated visual analysis features such as object detection or OCR.

In general, Azure Video Indexer provided the most comprehensive solution for video indexing within the scope of this thesis. Its built-in transcription and OCR capabilities can be used without extensive configuration, and the structured JSON output simplifies the implementation of the subsequent fact extraction pipeline.

3.4 Summary

This chapter reviewed the current state of research in automated fact-checking, with a focus on misinformation detection in videos, examined Information Extraction, and presented video tools suitable for processing raw video content.

Automated fact-checking has advanced considerably, especially for text-based claims using NLP methods to analyse text-based claims and retrieve supporting evidence. At the same time, misinformation increasingly spreads through multimodal content, especially videos that combine speech, images, and contextual metadata. Recent research has therefore extended fact-checking systems to multimodal settings, integrating textual, audio, and visual features, as well as user-generated metadata such as comments, to improve verification performance.

Structured Information Extraction is essential for converting unstructured video and text data into machine-readable formats. Recent research increasingly focuses on deep learning and large language models (LLMs), which offer scalable and flexible solutions for extracting structured information.

Video analysis tools support the structured preprocessing of raw video material. Services such as Microsoft Azure Video Indexer, Google Cloud Video Intelligence API, Amazon Rekognition Video, Clarifai, and Rev.ai provide functionalities including speech-to-text transcription, object detection, face recognition, and optical character recognition. These tools transform audiovisual content into machine-readable formats and generate timestamp-based metadata.

4 Conceptual Foundation and Design

This chapter presents the conceptual foundation and system design of the approach developed in this thesis. It first describes the underlying problem of extracting structured information from unstructured news video content and outlines the motivation for addressing this challenge. Based on these considerations, the system's functional and non-functional requirements are derived.

Building on these requirements, the chapter introduces the core concepts that guide the proposed fact extraction approach, including multimodal information extraction, entity-based fact modelling, and schema-guided fact representation. Finally, the conceptual system design is presented, describing the architecture of the processing pipeline that transforms news videos into structured fact reports.

4.1 Problem Description

News videos provide important information, but their unstructured content and combination of audio and visual elements make them not directly machine-readable. This format poses significant challenges for automated processing. Consequently, extracting and structuring information from news videos requires methods that handle and integrate these diverse data sources.

This section outlines the problem addressed in this thesis, highlights the underlying challenges, demonstrates the motivation for addressing them, and details the computational tasks required to transform unstructured news videos into a structured, machine-readable format.

4.1.1 Context and Motivation

As discussed in Chapter 2, news videos have become a key medium for sharing information in today's digital journalism. Unlike traditional text-based news, they combine spoken language, visual cues, textual overlays, and contextual metadata to convey detailed information about events [15]. Because news organisations primarily create these videos for human viewers, their unstructured, multimodal format poses major challenges for automated analysis, verification, and large-scale processing [3].

At the same time, the rapid growth of digital media and online news platforms has made it easier for misleading or conflicting information to spread [4]. Such conflicting information may arise from deliberate disinformation campaigns, journalistic errors, or differing interpretations of complex events and can spread quickly. As a result, this development highlights the need for automated systems capable of identifying, structuring, and organising conflicting information [9].

Automated fact-checking of news videos supports media monitoring by rapidly detecting inconsistencies in reporting. Furthermore, it integrates video-based information into knowledge bases and other verification applications. By extracting and structuring factual content, such systems reduce the effort required by human experts to analyse and verify key information within large volumes of video data [10].

Building on these considerations, this thesis aims to bridge the gap between unstructured news video content and structured fact representations used for comparison or verification.

4.1.2 Underlying Problem

In response to these challenges, this thesis addresses the task of transforming unstructured data from news videos into structured and semantically meaningful fact representations. A news video typically contains multiple types of information, including:

- Audio signals, including spoken language conveying narratives, statements, or opinions.
- Visual information, including textual overlays (e.g., titles, captions, and logos) as well as visual cues such as locations or the faces of publicly known individuals.
- Temporal structure, reflecting information distributed across the video timeline and requiring consideration of the order, duration, and co-occurrence of content to correctly structure information.
- Metadata describing attributes such as the source of the publication, the time of transmission, and the duration of the video.

Each type of information contributes to the content of a news video, but often appears redundant, fragmented, implicit, or noisy. As a result, direct extraction of facts is not feasible without intermediate steps of abstraction and integration. Therefore, the system must process a news video including its audiovisual content and metadata, and generate a structured report of key entities with their associated facts.

To achieve this, the system must identify relevant information units across multiple information types. Next, it must extract semantically meaningful facts, such as entity properties. Finally, the system must organise these facts into a machine-readable format suitable for further processing, such as automated fact-checking.

4.1.3 Task Abstraction

The task addressed in this thesis involves extracting information by combining techniques from video analysis, audio processing, and natural language processing. The system is required to perform the following key steps:

- Visual content analysis: Optical Character Recognition (OCR), which converts images of text into text data, is used to extract textual overlays such as headlines, speaker names, and captions from video frames.

- Audio content analysis: Automatic Speech Recognition (ASR), which converts spoken audio to written text, is used to transcribe spoken language for further processing.
- Textual analysis and NLP: Named Entity Recognition (NER) identifies key entities, including persons, organisations, and events. Information extraction and fact structuring convert unstructured textual information into standardised fact representations suitable for future comparison with external knowledge sources.
- Multimodal information integration: Information from audio and visual sources is aggregated to reduce redundancy and generate coherent, structured fact reports.

Integrating multiple information sources, resolving conflicts, and ensuring robustness against noisy or incomplete data present core challenges. When the system successfully addresses these challenges, it automatically extracts structured information from news videos, thereby supporting subsequent fact-checking and comparison tasks.

4.2 Requirements Specification

Based on these goals, the system requirements are derived directly from the research questions addressed in this thesis. Accordingly, the system is intended to:

- Enable automated extraction and structuring of information from unstructured news video content.
- Reduce informational complexity by focusing on key facts.
- Demonstrate technical feasibility through a working prototype.

The following sections specify the functional and non-functional requirements of the prototype developed in the course of this master's thesis.

4.2.1 Functional Requirements

The prototype is required to fulfil the following functional requirements:

- The prototype is required to support the analysis of news videos exclusively; other video genres are outside its scope.
- The prototype is required to allow users to upload news video files for analysis. The retrieval or download of videos from external platforms or content providers is out of scope.
- The prototype is required to enable users to initiate the fact extraction process for selected news video inputs.
- The prototype is required to extract textual information from spoken language in the audio track and from textual overlays detected in video frames.

- The prototype is required to identify and classify relevant entities within the extracted content.
- The prototype is required to aggregate and normalise extracted information to reduce redundancy and improve consistency.
- The prototype is required to extract properties associated with identified entities.
- The prototype is required to generate structured fact reports from the extracted information.
- The prototype is required to provide a mechanism to persist and export fact extraction results in a format suitable for distribution and evaluation.

4.2.2 Non-Functional Requirements

In addition to the functional requirements, the system is required to satisfy the following non-functional requirements, which influence both the architecture design and the conceptual processing model:

- **Performance**
  - The system is required to process news video inputs within a reasonable time frame, taking into account that the overall processing time depends on the duration of the video.
  - Time-consuming processing steps inherent to video and text analysis are acceptable, provided that the system remains responsive and usable.
- **Usability**
  - The system is required to provide an intuitive user interface for uploading news video files.
  - The system must indicate the processing and loading times resulting from computationally intensive analysis steps.
  - The system is required to present the results of the fact extraction process in a clear, comprehensible format.
- **Compatibility**
  - The system is required to support a predefined set of video file formats; other file types are not supported.
  - The system is required to be accessible via a web browser and optimised for use on non-mobile devices.
- **Software Quality**
  - The system is required to be built using a modular design, keeping different responsibilities separate.

- The system is required to ensure transparency of intermediate processing results to facilitate inspection and debugging.
- The system is required to be scalable to accommodate future extensions to the fact extraction pipeline.
- The system is required to be robust to noisy or imperfect input data.

4.3 Core Concepts

The following section describes the core concepts that underpin the proposed approach and address the research questions of this master's thesis by presenting the methodology for transforming unstructured news video content into structured, machine-readable representations. The section first covers the extraction of information from audiovisual content and metadata. Then, it outlines the organisation of facts around key entities and the use of schema-guided extraction to ensure consistency. Finally, it presents the design of a structured fact report that can be used for further processing.

4.3.1 Information Extraction from News Videos

Addressing the first research question requires transforming unstructured news video content into a format suitable for automated processing. News videos combine multiple information channels, including audio, visual content, and metadata, which must be analysed together to enable structured information extraction.

In this thesis, video content is decomposed into its information sources. The spoken language on the audio track is converted to text using Automatic Speech Recognition (ASR), enabling subsequent linguistic analysis. In parallel, Optical Character Recognition (OCR) is applied to video frames to extract textual overlays such as captions, speaker names, and headlines. These overlays often complement or clarify spoken content, particularly regarding entity names.

The extracted textual information is then combined with metadata, such as duration and source language. As described in Chapter 5, this combined data set serves as the foundation for subsequent NLP tasks, such as entity recognition and fact extraction.

By transforming multimodal video content into a unified textual representation, the system creates the basis for structuring previously unstructured news video data, directly addressing the first research question.

4.3.2 Entity-Based Fact Extraction

A central design decision in this thesis is to use entities as the basis for fact extraction, reducing complexity while preserving the essential content of news videos. Facts are organised around key entities and their associated properties, supporting the second research question by enabling structured, compact, and reusable representations of information derived from unstructured video data.

News videos frequently reference the same entities across different segments and modalities, including spoken language, textual overlays, and metadata. Organising information as entities allows the system to consolidate related information and reduce redundancy. Facts are represented as properties associated with entities, resulting in a consistent and machine-readable format suitable for further processing, comparison, and validation.

The entity types *person*, *organisation*, and *event* were chosen for the prototype. This choice reflects common news reporting content. News videos primarily describe events involving actors and institutions, with persons and organisations representing the main actors and events capturing the central occurrences of a report.

Focusing on these entity types limits the scope of extraction while retaining the most relevant information. Rather than extracting all details, the system focuses on key entities essential for fact analysis, enabling the transformation of unstructured video content into structured facts.

These entity types are well supported by existing NLP techniques. Methods such as Named Entity Recognition can reliably identify persons, organisations, and events in speech transcripts generated by Automatic Speech Recognition and textual overlays extracted via Optical Character Recognition.

In addition, persons, organisations, and events can be mapped to semantic models such as *schema.org*, enabling standardised fact representations and facilitating future comparison with external knowledge bases. Limiting the model to these three entity types keeps the scope manageable for this master's thesis while allowing additional types to be added in future work.

4.3.3 Schema-Based Fact Extraction

This thesis uses predefined schemas to guide fact extraction and organise information from news videos. These schemas define which properties may be extracted for each entity type, including *person*, *organisation*, and *event*, and provide a template for consistent output. By constraining the information extraction process, schemas reduce ambiguity, improve the reproducibility of results, and support their validation and comparison.

The system is based on the existing *schema.org* standard [60] to maintain consistency and compatibility. Properties from *schema.org* are used when possible, providing standardised, widely recognised descriptions of entities. However, the system is not strictly limited to *schema.org* properties. News video content often contains information that does not map directly to existing schema definitions, and enforcing strict compliance could lead to information loss or overly rigid representations.

Consequently, *schema.org* is used as a guiding reference rather than a hard constraint. When suitable properties exist, they are preferred in order to maintain comparability and clarity. When no appropriate property is available, additional properties are introduced to capture relevant information. This approach aims to balance standardisation and flexibility by enabling structured information extraction while remaining extensible to edge cases.

During fact extraction, the system first identifies which properties appear in the source material, then extracts their values. This ensures that the extracted facts are

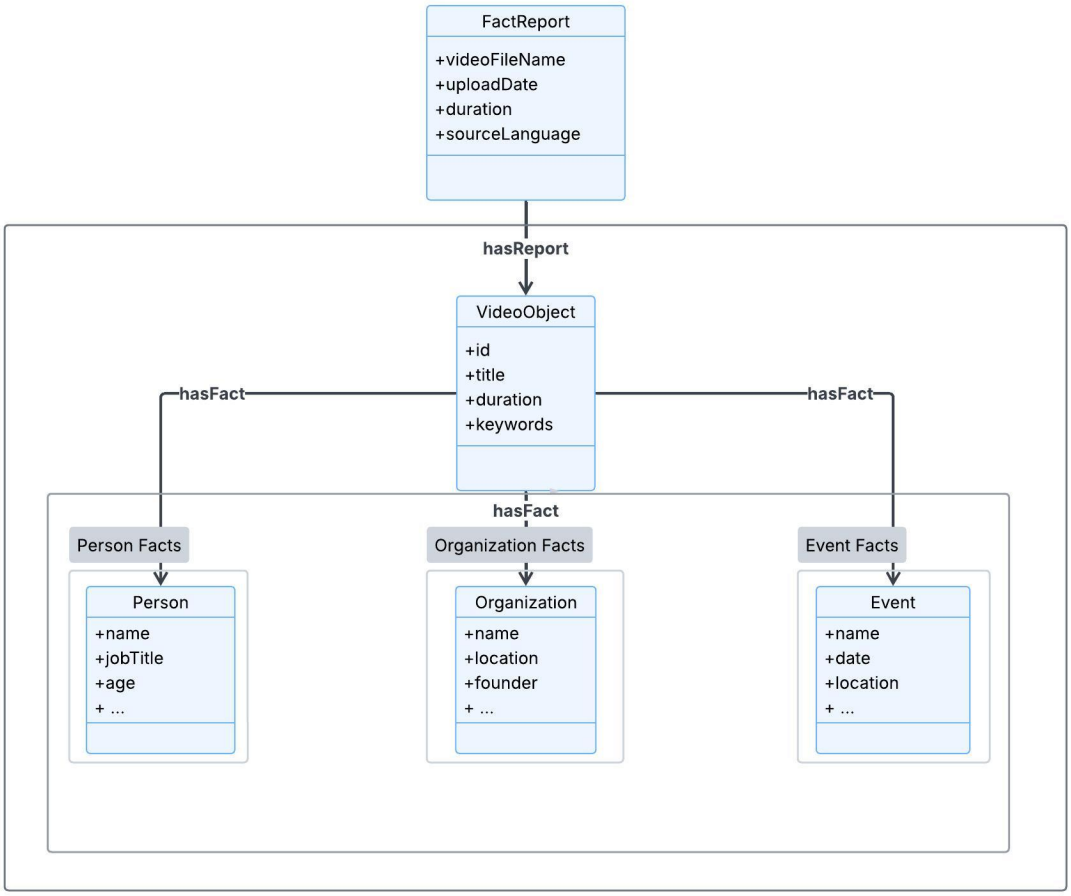


Figure 4.1: Fact report structure

clear, meaningful, and aligned with the conceptual model. The extraction strategy separates the identification of relevant information from its organisation, thereby creating a structured process for transforming unstructured news video content into reusable fact representations.

4.3.4 Fact Report Structure

During the conceptual phase of this master's thesis, a fact report structure was developed by manually analysing news videos and identifying a format suitable for representing their content in a structured, reusable manner.

The conceptual structure of the generated fact report is illustrated in Figure 4.1.

The fact report is represented as a structured and hierarchical document that captures both metadata about the analysed news video and the facts extracted from its content. The structure is designed to separate video-level context from extracted facts while

remaining extensible and machine-readable.

At the top level, the report contains global metadata describing the analysed video. This includes identifiers such as the video file name, upload timestamp, duration, and source language. These attributes provide contextual information and support traceability and comparison across different analyses.

The central element of the report is the *hasReport* field, which contains one or more video-specific report objects. Each report object represents the structured analysis of a single news item. It includes descriptive metadata such as title, duration, and keywords, as well as the extracted facts about key entities. Extracted facts are organised under the *hasFact* field and grouped into three entity categories: persons, organisations, and events. Each category contains a collection of fact entries derived from the video content if entities in that category are mentioned in the news video.

Each fact entry describes a single key entity and associates it with relevant properties such as names, dates, or locations. This representation allows the system to consolidate all information about a single entity, even if it appears across multiple segments of the video.

Overall, the fact report is structured in layers, from basic video metadata to analysed content and extracted facts. This organisation supports further automated processing, such as fact comparison and knowledge base integration, while remaining flexible enough for future extensions of the extraction pipeline.

4.4 System Design

This section describes the conceptual design of the system developed in this thesis. Building on the previously defined requirements and core concepts, the system design outlines how unstructured news video content is processed and transformed into structured fact representations.

The design follows a modular architecture that separates the main processing responsibilities into distinct components. At a conceptual level, the system consists of three main processing stages: video analysis, content processing, and fact structuring. These stages are organised as a sequential pipeline, where the output of one stage serves as the input for the next. The final result of the pipeline is a structured fact report that summarises the key entities and their associated properties extracted from the analysed news video.

4.4.1 System Architecture Overview

The system architecture is designed as a pipeline that transforms unstructured video content into structured fact representations through several processing stages. Each stage performs a specific task and contributes to the abstraction and structuring of the original video data.

The system processes a news video through the following main components:

- **Video Analysis Component:** The component extracts textual and metadata information from the audiovisual content of the video.

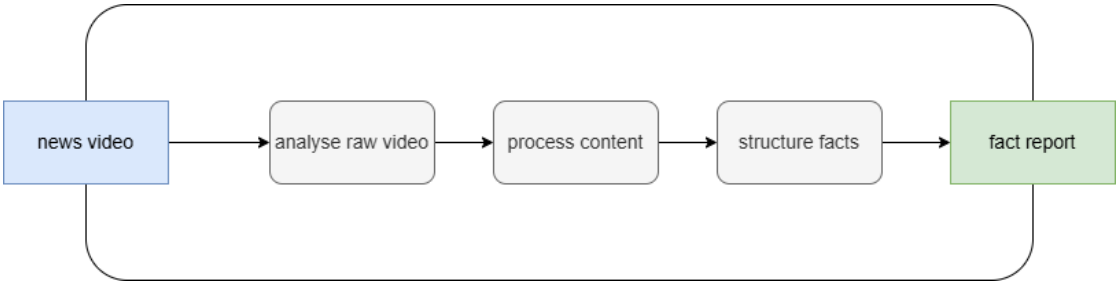


Figure 4.2: Fact extraction pipeline

- **Content Processing Component:** The component performs linguistic analysis and identifies relevant entities within the extracted text.
- **Fact Structuring Component:** The component converts the processed information into structured facts using predefined schemas.
- **Fact Report Generation:** The component organises the extracted facts into a structured report format for further analysis and evaluation.

The modular structure enables different processing steps to remain separated while allowing individual components to be extended or replaced in future work.

4.4.2 Processing Pipeline

Figure 4.2 provides a high-level overview of how a news video is transformed into a structured fact report.

The system processes news videos through a sequential pipeline consisting of several stages that progressively transform raw video data into structured information.

The workflow begins with the upload of a video file provided by the user. The video is then analysed to extract textual and metadata information from its audiovisual content. The extracted text is subsequently processed using Natural Language Processing techniques to identify relevant entities and reduce the information complexity of the video content.

In the next stage, the system performs fact structuring by identifying properties associated with the detected entities. These properties are organised according to predefined schemas that guide the structure of the resulting data. Finally, the extracted information is compiled into a structured fact report that represents the main content of the analysed news video.

The pipeline-based design enables the system to progressively refine the information extracted from the video by transforming its unstructured content into a structured machine-readable representation.

4.4.3 Video Analysis Component

The video analysis component is responsible for extracting textual and metadata information from the audiovisual content of a news video. Since the information contained in news videos is distributed across different channels, the system must analyse both the audio and visual components of the input video.

The audio track is processed using Automatic Speech Recognition to convert spoken language into a textual transcript. This transcript captures the content of the news report and provides the primary input for subsequent linguistic analysis.

In parallel, the video's visual component is analysed with Optical Character Recognition to detect textual overlays, such as captions, interview partners, or headlines. These elements often contain important contextual information, such as names, locations, or titles, that may not be explicitly mentioned in the spoken narration.

The extracted transcript and textual overlays are combined with basic video metadata, such as duration and language information. The resulting textual representation serves as the input for the subsequent content processing stage.

4.4.4 Content Processing Component

The content processing component performs linguistic analysis on the textual information generated during video analysis. Its primary goal is to identify relevant information units within the extracted text and prepare them for structured fact extraction.

The first step in this stage is to identify entities using Named Entity Recognition. This process detects persons, organisations, and events within the text. Because these entity types frequently represent the central actors and events in news reports, they serve as the basis for subsequent fact extraction. However, in future work, the system can be extended to integrate additional entity types.

After the key entities have been identified, the system aggregates and normalises the extracted information in order to reduce redundancy and improve consistency. Mentions of the same entity across different parts of the transcript or visual overlays are consolidated to create unified entity representations.

4.4.5 Fact Structuring Component

The fact structuring component transforms the processed textual information into structured, machine-readable fact representations. This stage focuses on extracting properties associated with the key entities identified during content processing.

This stage follows an approach in which predefined schemas specify the types of properties that may be extracted for each entity type. These schemas provide a structured template that ensures consistency and comparability across extracted facts.

For each entity identified in the previous stage, the system determines which relevant properties are explicitly mentioned in the source material. The corresponding property values are then extracted and organised according to the schema definitions.

This structured format enables the system to convert unstructured data into clearly defined facts associated with specific entities. As a result, the extracted information becomes suitable for further automated processing, comparison, or validation.

4.4.6 Fact Report Generation

The final stage of the system design focuses on organising the extracted facts into a structured fact report. This report is the system's primary output and provides a machine-readable summary of the analysed news video.

The report structure follows a hierarchical organisation. At the highest level, metadata describing the analysed video is recorded, including the video identifier, duration, and language. This contextual information ensures traceability and supports comparisons across different analyses.

In the report, extracted facts are grouped by the entity types defined in the conceptual model: persons, organisations, and events. Each entity is represented together with its associated properties, forming a structured description of the information contained in the news video.

The fact report supports both automated processing and human evaluation of the results.

4.4.7 User Interface

The system is designed to support a simple frontend that allows users to analyse news videos and retrieve structured fact reports. Users interact with the system through a graphical interface that provides access to the different stages of the processing pipeline.

The interaction begins with the upload of a news video file. After the video has been uploaded, the user can initiate the automated analysis and fact extraction process. The system then executes the processing pipeline and generates a structured fact report describing the key entities and their associated properties.

Once the analysis is completed, the user can view the extracted facts and export the generated report for further inspection or evaluation. The interface allows users to analyse individual news videos while maintaining transparency about the processing steps required to extract information.

4.5 Summary

This chapter introduced the conceptual foundation and system design for the automated extraction of facts from news videos. The problem of transforming multimodal and unstructured video content into structured representations was first analysed, followed by the definition of functional and non-functional system requirements.

Based on these requirements, the chapter presented the core concepts of the proposed approach, including multimodal information extraction, entity-based fact modelling, and schema-guided fact extraction. The conceptual system design then described a modular

processing pipeline consisting of video analysis, content processing, and fact structuring components.

The resulting design provides the foundation for the implementation of the prototype described in the following chapter.

5 Implementation

This chapter describes the implementation of the prototype developed to address the research questions in this thesis. It presents the system architecture, implementation decisions, and technologies used to transform unstructured news video data into structured fact data. The chapter covers both backend and frontend components, outlining how video analysis, content processing, and fact extraction are implemented and integrated into the prototype. In addition, it outlines the deployment setup and execution environment to support the reproducibility of the implemented approach.

5.1 Implementation Scope

This section outlines the scope of the implemented prototype. The implementation focuses on realising the core concepts required for the automated extraction of facts from news videos.

5.1.1 Objective of the Prototype

The primary objective of the implementation is to develop a functional prototype that demonstrates how unstructured data contained in news videos can be transformed into structured information. News videos contain multiple information sources, including spoken language in the audio track, textual overlays, and implicit semantic context. The prototype is designed to extract, aggregate, and structure these inputs into a unified representation that can be used for further analysis.

5.1.2 Relation to the Research Questions

The scope of implementation is defined by the three research questions addressed in this thesis.

First, the prototype addresses the challenge of structuring unstructured news video data by combining automated video indexing, transcription, Optical Character Recognition (OCR), and metadata extraction. The final fact structure provides access to individual information chunks, such as entities, keywords, and metadata, contained in the news video.

Second, the prototype implements mechanisms to compress and organise extracted information by focusing on key facts rather than all available information. This is achieved by selecting key information using Named Entity Recognition and fact extraction based on predefined fact schemas. By mapping information chunks to schemas for persons,

organisations, and events, the system reduces redundancy and emphasises the central aspects of the video content.

Third, the implementation demonstrates how the proposed news video fact-checking process can be realised as a working prototype. It integrates video analysis, content processing, and fact structuring into a single system that allows users to upload news videos, trigger automated analysis, and retrieve structured fact reports. The prototype thus serves as a proof of concept that validates the technical feasibility of the proposed approach.

5.1.3 Limitations

The scope of the implementation is limited to ensure feasibility for a master's thesis. The prototype focuses on a selected set of entity types, namely persons, organisations, and events, and relies on predefined schemas aligned with *schema.org* to structure extracted facts. The system is designed to process individual news videos and requires a single-user, local execution environment. Features such as fact comparison against external knowledge bases, support for multilingual news video, and large-scale deployment are outside the scope of this implementation. Furthermore, the quality of the extracted information depends on the accuracy and availability of external cloud services, as well as the behaviour of the employed language model.

5.1.4 Backend and Frontend Responsibilities

The implementation is divided into backend and frontend components with separate responsibilities. The backend implements the complete fact-checking pipeline, including video analysis, extraction and structuring of news video content, compression into key facts, and generation of structured fact reports. It also manages all interactions with external cloud services.

The frontend provides a step-based user interface that guides the user through the individual stages of the process, from video upload to fact visualisation. By keeping the processing logic separate from the user interface, the prototype remains modular, making it easier to evaluate the fact-checking workflow and the final result.

5.2 System Architecture

5.2.1 Overview of Architecture

In the following section, a component diagram and a sequence diagram of the prototype are presented to provide a high-level overview of the system architecture.

Component Diagram

Figure 5.1 illustrates the component diagram of the prototype, showing the internal structure of the main component, the *Video Fact Extractor*, and its interactions with external services.

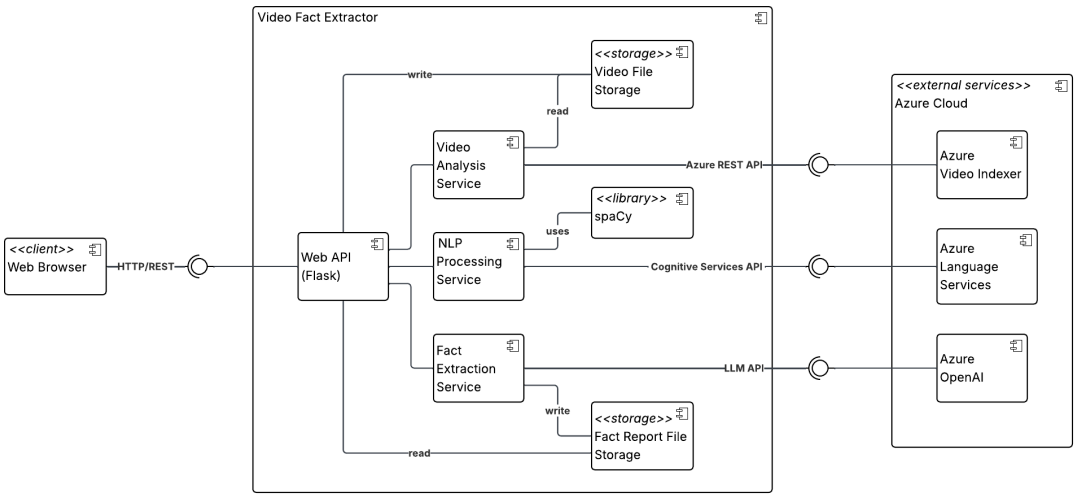


Figure 5.1: Component diagram

The prototype's interface is accessed via a REST Web API implemented with *Flask*, which a web browser uses to send client requests. Internally, the architecture comprises a *Video Analysis Service* that processes uploaded videos by accessing the *Video File Storage*, delegating advanced video analysis to the *Azure Video Indexer*. The extracted information is further processed by an *NLP Processing Service*, which performs linguistic analysis using the *spaCy* library and integrates *Azure Language Services*. Finally, the *Fact Extraction Service* uses a large language model via the *Azure OpenAI* API to extract facts, which are stored in the *Fact Report File Storage* and provided to the client through the Web API.

Sequence Diagram

Figure 5.2 illustrates the sequence diagram of the prototype, depicting the fact extraction workflow initiated by the user and executed by the system components.

The diagram shows how a user interacts with the system via a web service to upload a video, trigger processing and analysis, and request the final results, which are handled by the *Video Fact Extractor*. This central component coordinates multiple analysis engines, including the *Video Indexer*, *Azure Language Services*, *spaCy*, and an *AI Client*, to perform video indexing, Named Entity Recognition, summarisation, and fact extraction. Several processing steps, such as entity recognition and summarisation, are executed in parallel using both cloud services and local NLP components. The intermediate results, such as insights, summaries, and NER outputs, are stored locally for subsequent processing. Finally, extracted and validated facts are persisted and retrieved to generate a fact report, which is returned to the user.

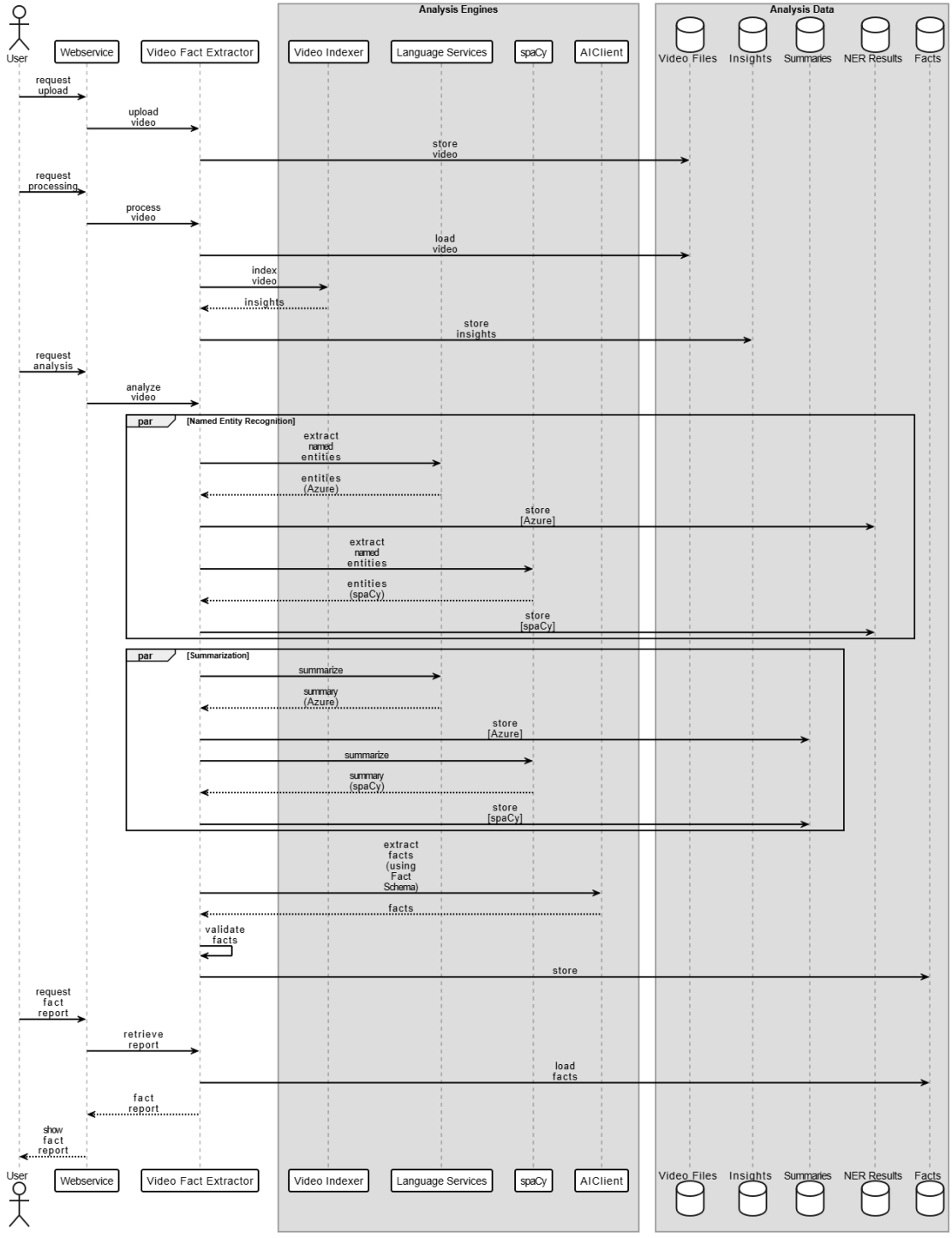


Figure 5.2: Sequence diagram

5.2.2 Technology Stack

The prototype is implemented in *Python 3.11*. Its technology stack was selected to support video analysis and advanced natural language processing. For that purpose, both cloud services and local libraries were evaluated to determine whether they meet the required specifications. Cloud services were ultimately chosen for computationally intensive tasks that would be inefficient to run locally due to resource constraints, while local components were kept for tasks that required more control and predictable costs. Commercial cloud providers such as *Microsoft Azure* typically charge per use, whereas open-source tools such as *spaCy* are generally free. These trade-offs helped determine whether to use cloud or local processing, especially for large or frequent tasks.

The backend of the prototype consists of several specialised components. The integrated tools are listed below.

- Video Processing:

  Microsoft Azure Video Indexer [35] is used for video analysis tasks such as transcription and insight extraction via a REST API.

- Named Entity Recognition:

  NER is implemented using two complementary approaches: *Azure Language Services* [39] for cloud-based entity extraction and the *spaCy* [38] library for local processing. This dual approach enables comparison of results and increases robustness by combining external and local NLP methods. Both systems are configured to extract the entity types *person*, *organisation*, and *event*.

- Summarisation:

  Similarly to NER, text summarisation is performed with both *Azure Language Services* [43] and *spaCy* [42], allowing the system to generate summaries using cloud-based models and locally executed algorithms. For both tools, an extractive method for summarisation is implemented. In this extractive method, parts of the text are extracted and stacked together to create a summary.

- Fact Structuring:

  The AIClient, powered by a large language model (LLM) via the *Azure OpenAI* API [46], is responsible for fact extraction based on predefined fact schemas. The deployed model is *GPT-4.1*, which has a knowledge cutoff of June 2024.

The frontend is implemented as a web user interface accessed through a standard web browser. Communication between the frontend and backend is handled via a REST API implemented using *Flask* [61]. This framework was chosen for its simplicity and flexibility, enabling users to upload videos, trigger processing steps, and retrieve generated fact reports.

5.3 Backend Implementation

5.3.1 Backend Architecture

The backend is implemented as a sequential video processing pipeline consisting of three primary components: video analysis, content processing, and fact structuring. Each component is implemented as a separate module and exchanges data with subsequent stages mainly through JSON outputs.

Video Analysis

The video analysis component handles the upload and automated analysis of video files using *Microsoft Azure Video Indexer*. This functionality is implemented in the `VideoIndexer` class.

Authentication with the Video Indexer API is performed using a subscription key and a time-limited access token. Since access tokens expire periodically, the implementation automatically renews tokens when unauthorised responses are encountered. Videos are uploaded using the advanced indexing preset, enabling the extraction of insights, including speech transcripts, keywords, source language, duration, and additional metadata.

As video indexing is an asynchronous process, the backend periodically polls the Video Indexer API until the processing state is marked as `Processed`. Once analysis is complete, the full indexing result is retrieved in JSON format and persisted locally. The stored result is used as input to subsequent processing steps.

Content Processing

The content processing component transforms the raw indexing output into structured textual and semantic representations suitable for the final fact extraction step. This component combines transcript processing, Named Entity Recognition, and text summarisation, and is implemented through the `InsightsProcessor`, `NerExtractor`, and `SummaryGenerator` classes.

The `InsightsProcessor` extracts and aggregates the full video transcript as well as Optical Character Recognition (OCR) results detected in video frames. In addition, it retrieves metadata, including the video name, upload date, total duration, report duration, source language, and keywords.

Named Entity Recognition is performed by the `NerExtractor` class, which combines the outputs of two complementary approaches. First, entities are extracted using *Azure Language Services*. Second, an additional extraction is performed using the *spaCy* language model. The results from both tools are merged and ordered by their textual offsets. Entities are categorised into persons, organisations, and events, and each entity is annotated with its source (transcript or OCR) and extraction tool (Azure or spaCy).

To improve entity quality and consistency, the implementation includes post-processing steps, including duplicate detection, category-based filtering, and optional spell-checking. Spell-checking is performed using phonetic similarity via the *Soundex* algorithm to merge variant spellings of the same entity. When resolving ambiguities, OCR-based mentions

are prioritised over transcript-based mentions, as names detected via OCR are more likely to be spelt correctly than those from the transcript. The combined NER results are persisted as a CSV file for reuse and evaluation.

To support completeness assessment and fact validation, a summarised version of the transcript is generated using the `SummaryGenerator` class. This component combines extractive summarisation from *Azure Cognitive Services* and *spaCy*. The final summary is created by merging non-duplicate sentences from both methods, resulting in a compact representation of the main content of the video.

Fact Structuring

The fact structuring component converts unstructured textual content and extracted entities into structured, machine-readable fact representations. This component integrates semantic analysis, schema-driven fact extraction, and output generation, and is implemented through the `AIClient` and `OutputGenerator` classes.

Semantic fact extraction is performed by the `AIClient` class, which accesses *Azure OpenAI*. Named entities identified during content processing are categorised into persons, organisations, and events. For each entity type, dedicated prompts are issued to the language model to identify and extract relevant properties from predefined JSON schemas for persons, organisations, and events, aligned with *schema.org* definitions. However, the prototype is not strictly bound to *schema.org*. If no relevant properties are available, non-conformant properties may be used.

The extraction process follows a two-stage interaction pattern. First, the model determines which entity properties are explicitly mentioned in the source text. Second, concise values are extracted for each identified property, strictly constrained to information contained within the transcript. Each entity type is processed individually. Persons and organisations are queried separately for each recognised instance, while events are processed collectively to handle multiple references to the same event. To ensure robustness, the implementation includes automatic validation of model responses and retry mechanisms for incomplete or malformed JSON outputs. A post-processing step removes underspecified or invalid facts, ensuring that only meaningful and well-defined information is retained.

The `OutputGenerator` class is responsible for consolidating metadata, extracted facts, and report information into structured output. Video metadata, including the file name, upload date, duration, and source language, is combined with one or more report objects that represent analysed video segments. In the evaluation data set, each analysed video contained only a single report object.

A report includes the video title, duration, extracted keywords, and a collection of facts grouped by entity type. The resulting structure is serialised as a JSON-LD file.

In addition to the JSON output, the backend generates a structured Excel report to support systematic evaluation and manual validation of extracted facts, which was used to evaluate the prototype. Separate worksheets are created for persons, organisations, and events, each including extracted properties with evaluation fields. An additional entity check worksheet lists entities derived from a summarised version of the transcript,

enabling completeness assessment. Finally, an evaluation worksheet aggregates validation statistics using automatically generated spreadsheet formulas.

5.3.2 Fact Schemas

The AI Client is provided with structured JSON schemas for the three main entity types: *person*, *organisation*, and *event*. These schemas define the expected data structure and ensure that extracted facts are consistent, machine-readable, and suitable for further processing.

Each entity type includes a general schema that specifies which properties can be extracted from the transcript, and a detailed schema that provides the corresponding property values when available. This two-level approach allows the system to track both the presence of information and the corresponding content.

Defining the person, organisation, and event schemas is necessary to provide the AI Client with a structured output template, improving comparability and consistency of the extracted facts. This approach ensures that facts are captured reproducibly.

5.3.3 Data Storage

All data generated and processed by the prototype is stored as locally persisted files. This includes several types of files, each generated at a different stage of the workflow:

- **Video files (`.mp4`):** The original source videos uploaded for analysis.
- **Video Indexer responses (`.json`):** The structured output from *Microsoft Azure Video Indexer*, containing transcripts, keywords, and other extracted insights.
- **NER results (`.txt`):** Named Entity Recognition outputs generated during content processing, listing all identified persons, organisations, and events.
- **Final fact outputs (`.json`):** Structured representations of extracted facts, including video metadata, report details, and information about persons, organisations, and events mentioned in the video.

5.3.4 Configuration and Environment Variables

The prototype utilises a `.env` file to securely store and manage configuration information. This file contains credentials, endpoints, and identifiers required by the services used in the workflow, including:

- **Azure Video Indexer:** API key, account ID, and location for video upload and analysis.
- **Azure OpenAI Client:** Subscription key, deployment name, API version, and endpoint information for AI fact extraction.
- **Azure Cognitive Services:** API key and endpoint URL for Named Entity Recognition and text summarisation.

5.4 Frontend Implementation

This section describes the implementation of the prototype's frontend. It presents the frontend architecture, discusses the user interface design choices, and explains the data flow between the frontend and the backend.

5.4.1 Frontend Architecture

Architecture Overview

The frontend of the prototype is implemented as a web application using the Flask framework. Flask serves the HTML page, provides HTTP endpoints, and connects the user interface to the backend components that implement the fact extraction pipeline for news videos. The application follows a server-rendered architecture, where the main view is generated on the server, while dynamic behaviour and user interactions are handled on the client using JavaScript.

The central entry point of the frontend is the `index.html` template, which is rendered when the root endpoint of the application is accessed. This template defines the overall structure of the user interface, including a step-by-step visualisation of the news video processing pipeline, a sidebar that presents the interaction options for each step, and an area for displaying the fact extraction results.

Frontend-Backend Communication

The client-side interaction is handled by the JavaScript file `script.js`, which tracks the currently selected workflow step and updates the interface accordingly. The sidebar displays a textual description, relevant input elements, and an action button for each processing step when the corresponding step is selected on the main page. User actions, such as selecting files, dragging and dropping videos, or clicking buttons, are sent to the backend via asynchronous HTTP requests, as the individual processing steps handled by the backend can be time-consuming.

Communication between the frontend and the backend is realised through a set of REST endpoints. Video files are uploaded via an endpoint using multipart form data, then stored on the server in the `static/files` directory. For subsequent processing steps, the frontend retrieves lists of available videos by sending GET requests to the corresponding endpoints. These videos are then presented to the user in a drop-down selection field. POST requests trigger video processing, content analysis, and fact visualisation.

In addition, the frontend communicates with the backend via structured JSON responses that provide information about uploaded videos, processed and analysed video files, and extracted fact data. JSON is used to deliver lists of available videos for each processing step, confirm the completion of operations such as video upload, processing, and content analysis, and transmit the fact extraction results that are displayed in the user interface. This approach allows the frontend to remain independent of the backend's internal processing logic, as all necessary data is communicated in a standardised format.

Visualisation

The visual layout is defined using custom Cascading Style Sheets (CSS) stored in the file `styles.css`. The main view is designed as a pipeline using a grid layout that represents the sequential structure of the fact extraction process. Individual workflow steps are displayed as colour-coded, interactive elements with hover effects to indicate interactivity. A sidebar component presents step-specific information and input elements. This occurs when the user clicks a step button in the main view.

5.4.2 User Interface Design

This section describes the user interface of the prototype developed and explains its main functional components. The interface supports the execution of the individual steps of the video analysis pipeline and provides access to the corresponding input and visualisation elements. The following subsections describe the structure of the main screen and detail each processing step.

Main Screen

The application is organised as a step-based workflow that guides the user through the fact extraction pipeline. The main interface presents four sequential stages: video upload, video processing, content analysis, and fact visualisation. Each stage is represented by a clickable button on the main screen. When a button is selected, a sidebar opens on the left side of the interface, providing a brief description of the corresponding step and the relevant input elements, such as a drag-and-drop area or a drop-down menu for video selection.

Although all four stages must be completed to perform a full analysis of a news video, the system allows individual stages to be executed independently, enabling the user to process different videos at intermediate steps if required. Figure 5.3 illustrates the main interface of the application.

Upload Video

The video upload step is the first stage of the processing sequence. By clicking the grey button labelled *1. Upload Video*, a sidebar opens on the left side of the screen and displays the corresponding interface elements. During this step, a user-defined video file is uploaded to the web server. Once the upload is completed, the video becomes available for subsequent processing stages.

Users can upload a video either by dragging and dropping the file into the designated area or by clicking the drag-and-drop section, which opens a file selection dialogue allowing a video to be chosen from the local device. After selecting a video, the upload is initiated by clicking the upload button located below the drag-and-drop field. Figure 5.4 shows a screenshot of the upload section.

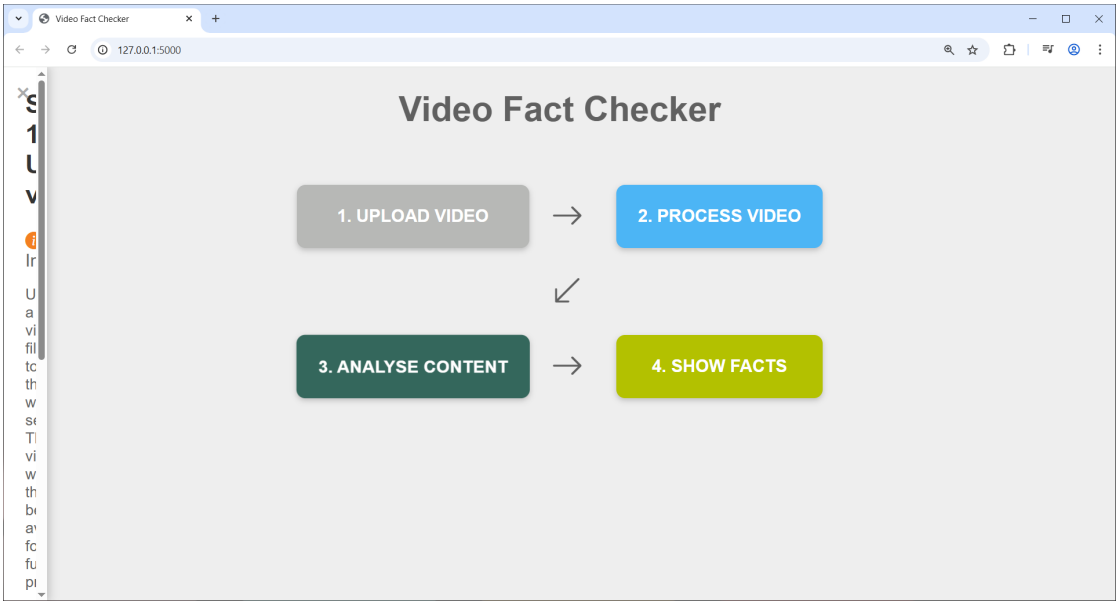


Figure 5.3: User interface

Process Video

The next step in the sequence shown on the main page of the user interface is processing a previously uploaded video. To initiate this step, the user clicks the blue button labelled *2. Process Video*. Subsequently, a sidebar appears on the left side of the screen, displaying the interface elements associated with the processing step. Figure 5.5 shows a screenshot of the layout for this section.

To select a video for processing, the user can choose from a list of previously uploaded videos using a drop-down selection field, as displayed in Figure 5.6. Figure 5.7 shows a screenshot in which the user has selected a video. After selecting a video, the processing step is initiated by clicking the process button located below the drop-down field.

Analyse Content

The third step in the fact extraction sequence is to analyse the content of a previously uploaded and processed video. This step is initiated by clicking the blue button labelled *3. Analyse Content*. When the sidebar opens on the left side of the screen, it displays a drop-down field similar to the one used in the video processing step.

The user can select a previously uploaded and processed video from the drop-down field and confirm the selection by clicking the *Analyse* button located below it. Figure 5.8 shows a screenshot of the content analysis section.

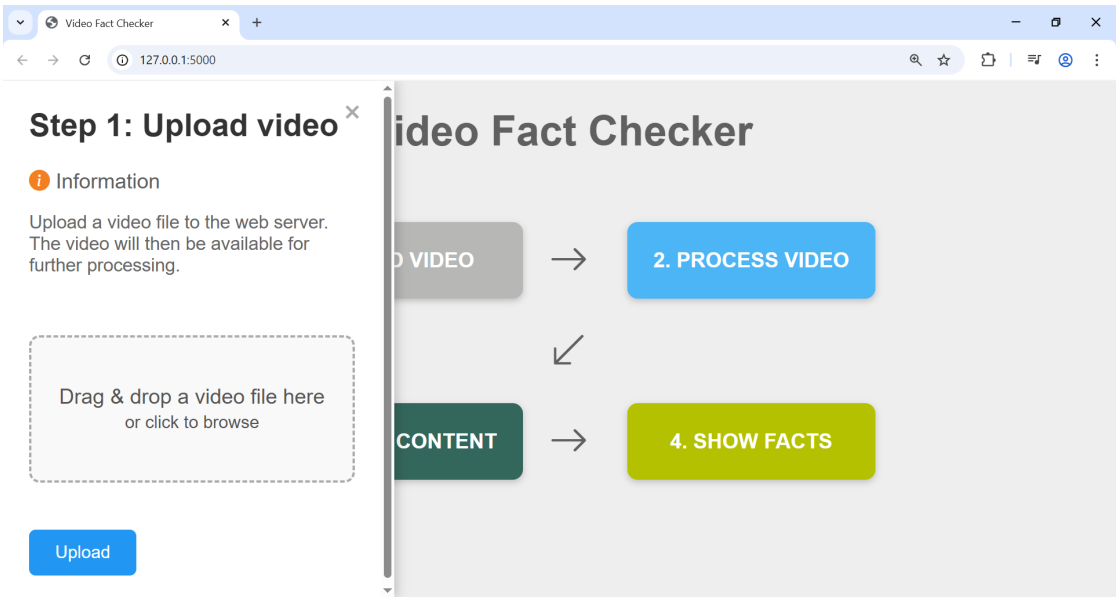


Figure 5.4: User interface: upload video

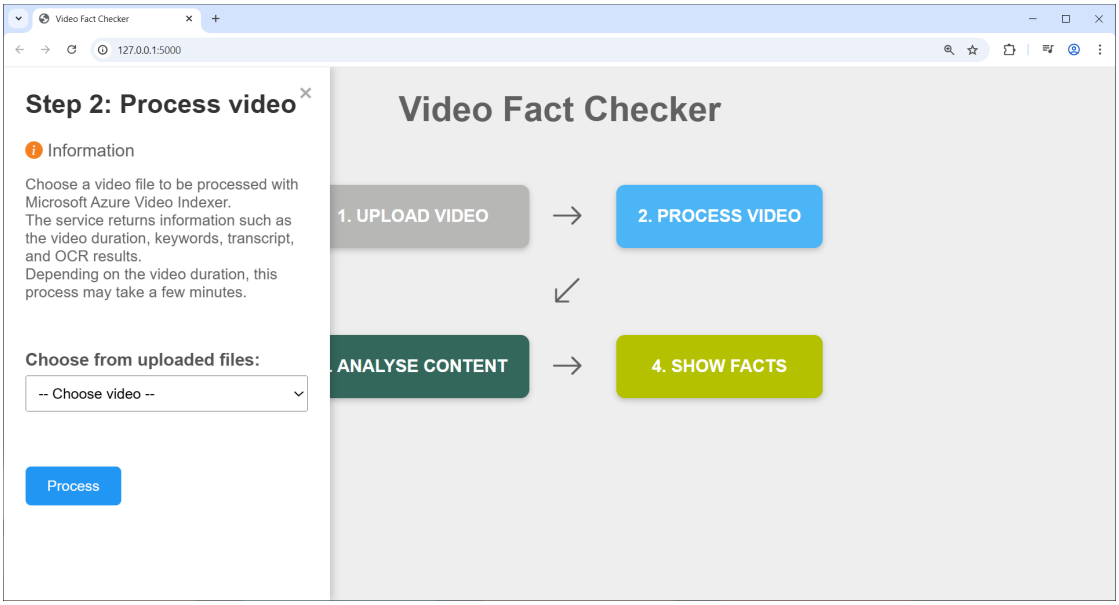


Figure 5.5: User interface: process video

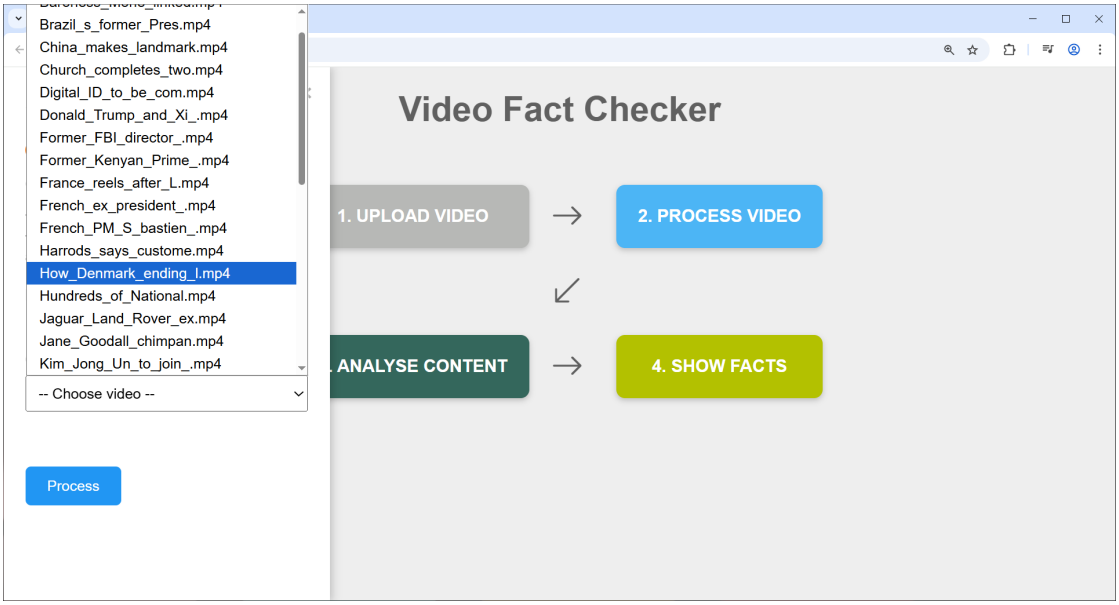


Figure 5.6: Process video: video selection

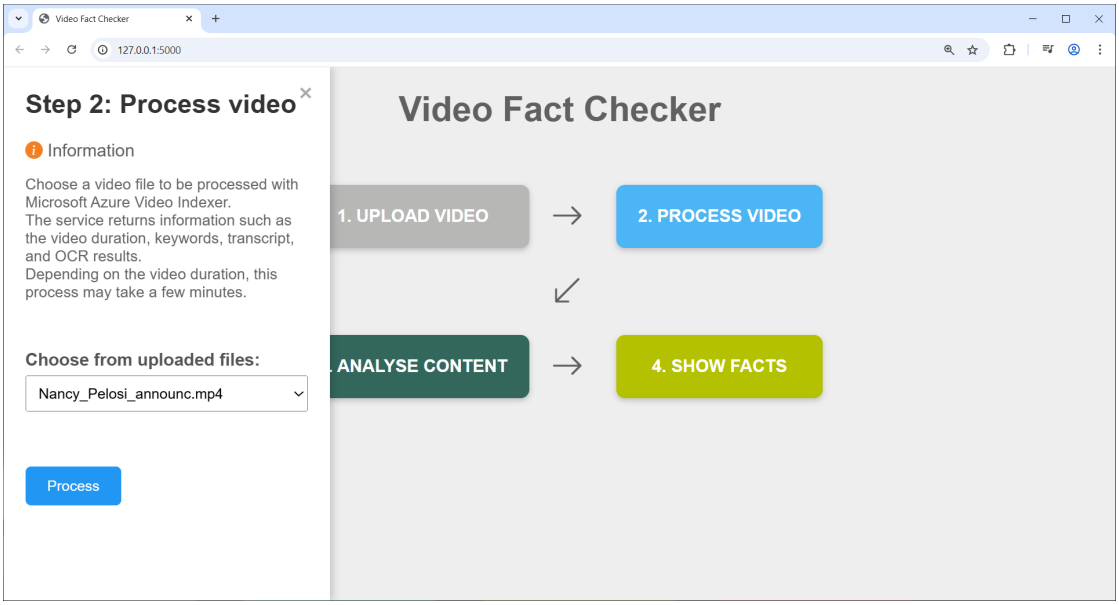


Figure 5.7: Process video: selected video

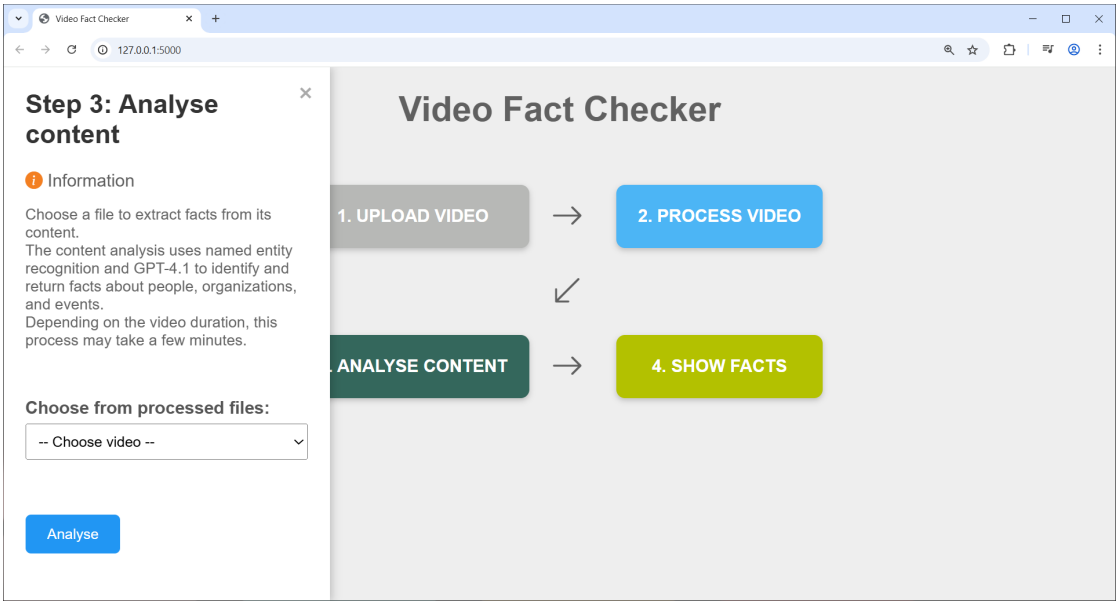


Figure 5.8: User interface: analyse content

Show Facts

The final stage of the workflow focuses on visualising the results of fact extraction for a previously uploaded, processed, and analysed video. Figure 5.9 shows a screenshot of the layout for this section. By clicking the green button labelled *4. Show Facts*, a sidebar containing a drop-down selection field is opened. The user can then select a video from the drop-down list and confirm the selection by clicking the *Show* button located below the field.

Once the video selection has been confirmed, the fact visualisation area is displayed at the bottom of the main page. This section presents the fact report generated for the processed news video, structured in subsections for metadata, report data, and fact data, as shown in Figure 5.10. Each subsection can be expanded, as demonstrated in Figure 5.11. The section containing the facts is divided into three main categories: person(s), organisation (s), and event(s), where facts about each entity type are listed (see Figure 5.12).

In addition, the fact report can be downloaded by clicking the download button located in the upper-left corner of the fact visualisation section. Upon user request, a download action is triggered that retrieves an automatically generated Excel file from the backend and initiates the download via the browser.

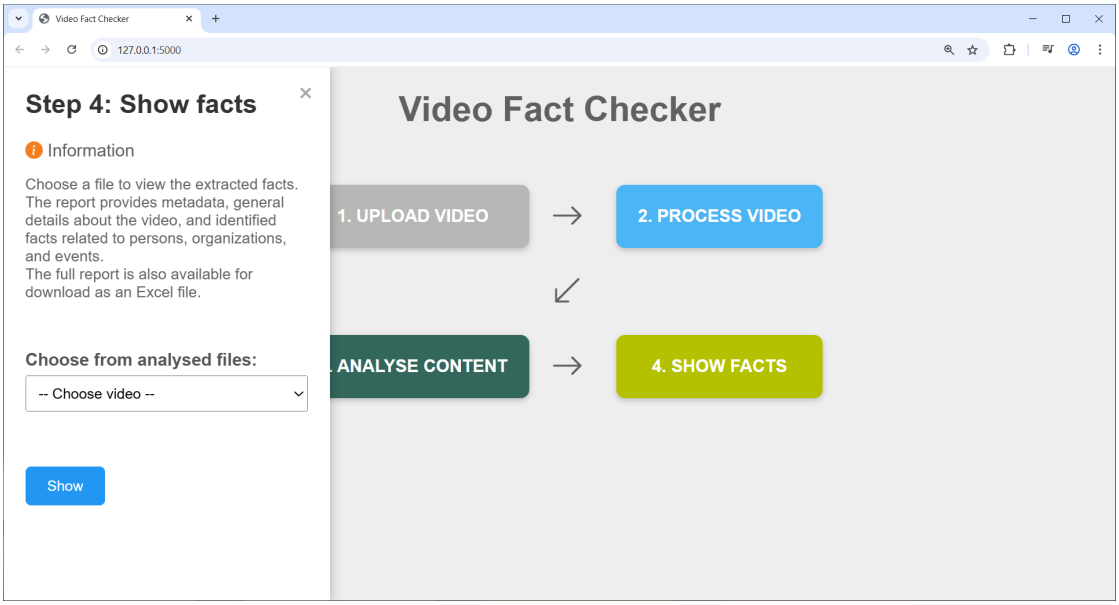


Figure 5.9: User interface: show facts

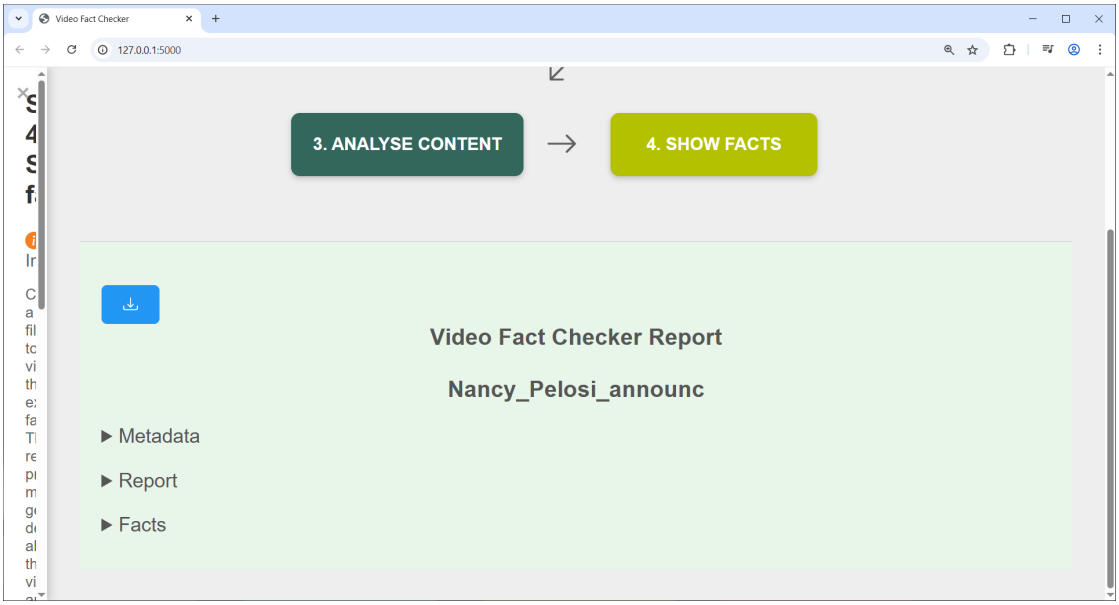


Figure 5.10: Show facts: fact report

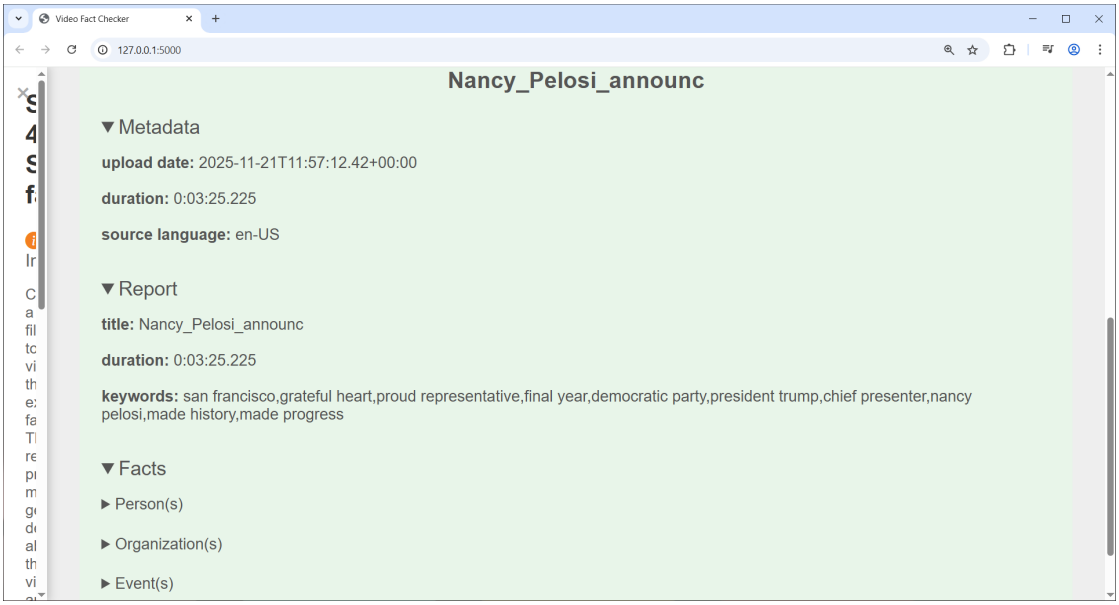


Figure 5.11: Show facts: facts



Figure 5.12: Show facts: person facts

5.5 Deployment and Execution

This section describes the prerequisites, setup, and execution of the implemented prototype. It outlines the required software and external services, explains how to obtain and configure the source code, and describes how to execute the system in a local environment.

Prerequisites

The following prerequisites must be fulfilled in order to deploy and execute the system:

- Python version 3.11 or later
- A Microsoft Azure Video Indexer account
- A Microsoft Azure Cognitive Services account
- A Microsoft Azure OpenAI account

Source Code Repository

The complete source code of the prototype is available in a Git repository and can be retrieved by cloning the following URL: `https://git01lab.cs.univie.ac.at/thesis/2023ws/01268478_kapeller_angelika.git`

Dependency Installation

All required Python dependencies are specified in the file `requirements.txt`. The dependencies can be installed using the Python package manager `pip` after setting up a suitable environment.

Configuration

Access credentials for the Microsoft Azure services are required for execution. These credentials include API keys and service endpoints for video indexing, content analysis, and fact extraction. The system assumes that all required credentials are provided via environment variables at runtime. No credentials are stored directly in the source code.

Execution

After installing the dependencies and configuring the required environment variables, the system can be started by executing the `main.py` script. This launches the web application that provides the user interface and exposes the backend endpoints. Once the application is running, the user interface can be accessed via a web browser at: `http://127.0.0.1:5000`.

Execution Environment

The prototype was developed and tested on a local workstation. The deployment assumes a single-machine setup and does not require containerisation or distributed execution.

Runtime Considerations

The runtime of individual processing steps depends on the length of the input videos and the response time of the external Azure services. As a result, steps such as video processing, content analysis, and fact extraction may take several minutes to complete.

5.6 Summary

The implemented prototype demonstrates an approach to automatically extracting facts from news videos. The system integrates video analysis, natural language processing, and fact extraction. Unstructured data from videos, including spoken language, textual overlays, and metadata, is processed, aggregated, and transformed into a structured fact representation for further analysis and validation.

The prototype is divided into backend and frontend components. The backend implements the fact extraction pipeline, which includes video analysis, content processing using Named Entity Recognition, and fact structuring via a large language model guided by predefined fact schemas. The prototype integrates cloud services and local libraries to improve performance while maintaining flexibility. The frontend provides a step-based web interface for uploading videos, triggering processing, and visualising extracted facts, communicating with the backend through REST endpoints and structured JSON responses.

Data is stored locally at each stage of the workflow, including uploaded video files, intermediate analysis outputs, NER results, and final structured fact reports in JSON and Excel formats.

The system implements the key functions relevant to the research questions: it structures news video content into accessible information chunks, compresses and organises it into key facts, and provides a working prototype for automated fact extraction from news videos. Limitations of the current implementation include support for only a limited set of entity types (persons, organisations, and events), operation in a single-user, local environment, and dependency on external cloud services for video analysis and natural language processing.

6 Evaluation

The evaluation section presents the methodology used to assess the prototype for this master's thesis. Then it reports the test results and findings. The objective is to examine the prototype's performance in extracting facts from news videos and to identify its strengths and weaknesses.

This chapter first provides an overview of the evaluation data set, followed by a description of the procedures applied to process news videos using the fact extraction prototype. Then it outlines the criteria used to assess the extracted results. Subsequently, the evaluation results are presented, with a focus on overall system performance and on identifying recurring errors in the automatic fact extraction process. Based on these findings, potential improvements to the implementation are discussed, highlighting how they could enhance fact extraction from news videos in future work.

6.1 Evaluation Data

6.1.1 Selection Criteria

For the prototype evaluation, 31 videos were selected that were characterised by the following criteria:

- The videos belong to the genre of news videos.
- The source language of all videos is English.
- The videos were published by the BBC (British Broadcasting Corporation).
- Each video represents a single news report.
- The videos were published between October 2025 and November 2025.
- The videos are in the mp4 file format.

6.1.2 Data Set

The data set used for the prototype evaluation consists of 31 news videos, all published by the BBC (British Broadcasting Corporation). The collection spans a broad range of topics, including global political developments, legal and criminal cases, cybersecurity incidents, climate and environmental crises, technological and medical advances, social issues, and notable cultural and leadership events. Table 6.1 lists the evaluated videos.

No.	Video title	URL	Access date
1	Madagascar President Andry Rajoelina hiding in 'safe place' as he warns of coup attempt	https://www.youtube.com/watch?v=nYzSd2v5pAU	15.10.2025
2	Hundreds of National Guard troops arrive in Illinois, US	https://www.youtube.com/watch?v=f6r1nM2I0ws	15.10.2025
3	Digital ID to be compulsory for all UK adults under government plans	https://www.youtube.com/watch?v=mb6jr16VtLE	28.09.2025
4	Former FBI director James Comey indicted on two charges	https://www.youtube.com/watch?v=34OeaWFxa1E	28.09.2025
5	Harrods says customers' data stolen in IT breach	https://www.youtube.com/watch?v=XzVboKJcD48	28.09.2025
6	Jaguar Land Rover extends shutdown after cyber-attack	https://www.youtube.com/watch?v=Ygcvh3hKNds	28.09.2025
7	Kim Jong Un to join Putin and other leaders at China military parade	https://www.youtube.com/watch?v=iCDEZsVPTNs	28.09.2025
8	Spain counts the cost of 'nightmare' wildfire summer	https://www.youtube.com/watch?v=4p4_eL8WeP4	30.09.2025
9	Mount Everest rescuers battle heavy snow with hundreds still stranded	https://www.youtube.com/watch?v=to4HbvbAbH8	15.10.2025
10	Sanae Takaichi to become first female Prime Minister of Japan	https://www.youtube.com/watch?v=DceviWlgaZM	15.10.2025
11	Jane Goodall, chimpanzee expert and animal rights campaigner, dies age 91	https://www.youtube.com/watch?v=NLyfswruKSg	15.10.2025
12	Baroness Mone-linked company ordered to repay UK government £122m over PPE contract	https://www.youtube.com/watch?v=JQ0PcKoaf-U	15.10.2025
13	Paralympic athlete challenging big brands over selling single shoes	https://www.youtube.com/watch?v=RWkFIYQSKPA	15.10.2025

No.	Video title	URL	Access date
14	Sarah Mullally named as new Archbishop of Canterbury	https://www.youtube.com/watch?v=FUa1kD58FK8	15.10.2025
15	Taliban imposes internet shutdown in Afghanistan	https://www.youtube.com/watch?v=k9If5mumQ8g	15.10.2025
16	Donald Trump and Xi Jinping discuss future of TikTok in US	https://www.youtube.com/watch?v=rR3mjjQS_NO	15.10.2025
17	Brazil's former President Jair Bolsonaro found guilty of plotting military coup	https://www.youtube.com/watch?v=l_DKHPUF2yU	15.10.2025
18	Church completes two-day journey across Swedish city	https://www.youtube.com/watch?v=3VGSkDOgzFo	15.10.2025
19	No immediate plans for Trump-Putin meeting, White House official says	https://www.youtube.com/watch?v=CDSLzAfGJqE	22.10.2025
20	French ex-president Nicolas Sarkozy goes to jail	https://www.youtube.com/watch?v=IEWMOqQs2pU	22.10.2025
21	France reels after Louvre jewellery heist	https://www.youtube.com/watch?v=jes_hmeOqqI	22.10.2025
22	Life-changing eye implant helps blind patients read again	https://www.youtube.com/watch?v=p-_d8bAyXXM	22.10.2025
23	French PM Sébastien Lecornu survives no-confidence votes	https://www.youtube.com/watch?v=imjTWAdgK98	22.10.2025
24	Thousands in UK sue Johnson & Johnson over talcum powder cancer risks	https://www.youtube.com/watch?v=Kis5lvbrJwc	22.10.2025
25	Former Kenyan Prime Minister Raila Odinga dies aged 80	https://www.youtube.com/watch?v=wgSws1_bAoo	22.10.2025
26	Peru to declare state of emergency over protests	https://www.youtube.com/watch?v=spGwUMT35NA	22.10.2025

No.	Video title	URL	Access date
27	China makes landmark pledge to cut its climate emissions	https://www.youtube.com/watch?v=Cl8VekAJpAo	15.10.2025
28	Mexico's Fatima Bosch crowned Miss Universe	https://www.youtube.com/watch?v=L_BnuNChE-k	21.11.2025
29	US Congress votes to send Jeffrey Epstein files bill to Donald Trump	https://www.youtube.com/watch?v=TNR2XD9WXhM	21.11.2025
30	Super Typhoon Fung-wong makes landfall in Philippines	https://www.youtube.com/watch?v=c9MMmzyMQkI	21.11.2025
31	Nancy Pelosi announces retirement after decades in US Congress	https://www.youtube.com/watch?v=USCotn9WVyO	21.11.2025

Table 6.1: Data set

6.1.3 Limitations

Since the data set represents only a small subset of potential news videos, the following limitations should be considered:

- All news videos are in English; other languages were not included as the prototype was designed to support fact extraction only in English.
- All news videos were published by the same broadcasting company; other broadcasters were not included in the evaluation.
- The prototype focuses on extracting facts about persons, organisations, and events. Other entities present in the videos were not analysed.
- The content of the videos was not used as a specific selection criterion, resulting in a broad range of topics. However, some subject areas may be overrepresented, while others may be underrepresented in the data set.
- The corpus of news videos was created specifically for this master's thesis. Manual selection may have introduced a bias in the data set's composition.

6.2 Automatic Fact Extraction

To evaluate the news videos, the proposed fact extraction prototype was employed. The following section describes the preparations taken and the procedure used for the automatic extraction of facts from news videos. Furthermore, it provides an overview of the structure and content of the fact report automatically generated by the prototype.

6.2.1 Preparation and Requirements

Before starting the automatic fact extraction process using the prototype, the following preparations were required:

- A data set consisting of 31 news videos was selected.
- The news videos were made available as downloaded video files.
- The video files were converted to the mp4 format.
- The prototype was deployed and made accessible via a web service.

6.2.2 Fact Extraction by the Prototype

After the data set was collected, the 31 news videos were processed using the prototype. The objective was to automatically generate a fact report for each video, which could then be manually validated. To achieve this, each video was uploaded to the prototype, processed and analysed, and the resulting output was downloaded as a fact report. The following subsections describe each step of this process in detail.

Upload Video

In the first step, each news video had to be uploaded to the prototype via the web service interface. For this purpose, the video file had to be available locally. Once the video file had been successfully uploaded to the prototype, it could be further processed.

Process Video

The second step of the automatic fact extraction process involved processing the news video file using *Microsoft Azure Video Indexer*. As described in Chapter 5, the video file was transmitted to the Video Indexer API, where several processing tasks were performed, including transcript generation, Optical Character Recognition (OCR), and keyword identification. The API then returned the extracted insights to the prototype, transforming the raw video file into a structured representation that was required for fact extraction in the subsequent analysis. Once these insights were available, the news video was ready for detailed content analysis.

Analyze Content

In the third step, the content of the news video was analysed to extract facts and organise them into a standardised fact structure. Therefore, the information obtained from the previous processing step was used to extract facts about persons, organisations, and events mentioned in the news video. This process began by identifying these entities using Named Entity Recognition (NER). Next, information about the identified entities was collected by an AI Client that aimed at summarising the available information. The

result was structured into a standardised fact format. Finally, the extracted information was validated and checked for potential inaccuracies.

Show Facts

The final step involved accessing the facts extracted from the news video. Therefore, the prototype provided a structured, readable view of the identified persons, organisations, and events, along with their associated properties. In addition, a fact report containing all extracted facts was downloaded. This fact report was also used to perform the manual analysis of the news videos and to compare the results of the prototype's automatic analysis with the manual evaluation.

6.2.3 Fact Report

The fact report is provided as an Excel file consisting of several worksheets that contain the following information:

- **Video Info:** This worksheet contains general metadata about the video, including the file name, upload date, duration, and source language.
- **Persons:** This worksheet compiles the extracted facts for each identified person. Each row corresponds to a single person, while the columns list the associated properties (e.g., name, age, and occupation). For each property, an additional evaluation column is provided for manual validation. An example of the fact report worksheet for persons is demonstrated in Figure 6.1.

	A	B	C	D	E
1	id	name	name - evaluation	jobTitle	jobTitle - evaluation
2	Nancy_Pelosi_announc_person_1	Nancy Pelosi	1	U.S. House Speaker, Representative for San Francisco	1
3	Nancy_Pelosi_announc_person_2	Barack Obama	1		
4	Nancy_Pelosi_announc_person_3	Donald Trump	1	Former President of the United States	0

Figure 6.1: Fact report: persons

- **Organizations:** This worksheet follows the same schema as the *Persons* worksheet. It compiles the extracted facts for each identified organisation, with each row representing a single organisation and the columns listing the associated properties (e.g., name, founder, and location). For each property, an additional evaluation column is provided for manual validation.
- **Events:** This worksheet follows the same schema as the *Persons* and *Organizations* worksheets. It compiles the extracted facts for each identified event, with each row corresponding to a single event and the columns listing the associated properties (e.g., name, date, and location). For each property, an additional evaluation column is provided for manual validation.

- **Entity check:** This worksheet is used for manual validation. It lists the entities identified through named entity recognition applied to the video summary and serves as a reference for verifying whether relevant entities are missing from the automatically extracted facts. The entities derived from the summary are used because they are assumed to represent the key elements of the entire video.

- **Evaluation:** This worksheet summarizes the evaluation results for the *Persons*, *Organizations*, and *Events* worksheets. An example of the evaluation worksheet is demonstrated in Figure 6.2. Each row corresponds to a specific property of an entity type (person, organisation, or event). The columns indicating true and false results contain the number of cases in which a property was correctly or incorrectly identified during manual validation. The example in Figure 6.2 indicates, in the second row, that for the video identified as *Jane_ Goodall_ chimpan*, properties of the type *name* associated with *person* entities were correctly detected in two cases and were not falsely detected in any case.

	A	B	C	D	E
1	**video**	**entity**	**property**	**true**	**false**
2	Jane_Goodall_chimpan	person	name	2	0
3	Jane_Goodall_chimpan	person	jobTitle	1	0
4	Jane_Goodall_chimpan	person	deathDate	2	0
5	Jane_Goodall_chimpan	person	description	2	0
6	Jane_Goodall_chimpan	person	affiliation	1	0
7	Jane_Goodall_chimpan	person	nationality	1	0
8	Jane_Goodall_chimpan	person	worksFor	1	0
9	Jane_Goodall_chimpan	person	url	0	1
10	Jane_Goodall_chimpan	person	birthPlace	1	0
11	Jane_Goodall_chimpan	organization	name	2	0
12	Jane_Goodall_chimpan	organization	description	2	0
13	Jane_Goodall_chimpan	event	name	3	0
14	Jane_Goodall_chimpan	event	location	3	0
15	Jane_Goodall_chimpan	event	date	1	2
16	Jane_Goodall_chimpan	event	result	3	0

Figure 6.2: Fact report: evaluation

6.3 Manual Fact Validation

After the automatic fact extraction was performed on the data set, the extracted results were manually validated. This section describes the manual validation procedure and the evaluation metrics applied.

6.3.1 Evaluation Procedure

The manual validation process began with watching the corresponding news video. Subsequently, the generated fact report was examined manually. For this purpose, the *Persons*, *Organizations*, and *Events* worksheets were reviewed to assess whether the extracted properties had been correctly identified.

In addition, the *Entity Check* worksheet was used to verify whether any relevant entities were missing from the fact report. As described previously, the entities identified in the video summary were used for this step, as they are assumed to represent the key entities of the full news video.

6.3.2 Evaluation Metrics

The evaluation labels in the *Persons*, *Organizations*, and *Events* worksheets indicate whether an extracted property is correct or incorrect. These labels do not assess the factual truth of the information in the sense of fact verification. Instead, a result is considered correct if the information is explicitly mentioned in the news video and incorrect if it is not. A label of 1 denotes a correct result, whereas a label of 0 indicates an incorrect result, corresponding to a hallucinated fact.

In the *Entity Check* worksheet, a label of 1 indicates that the entity is present in the fact report, whereas a label of 0 indicates that the entity is missing. One possible reason for the label 0 is that the prototype failed to detect the entity. Another possibility is that the entity was initially detected but later removed. This occurs when it is associated with fewer than two properties and is therefore considered unsuitable for subsequent fact comparison tasks.

6.4 Results

This section presents the results of the automatic fact extraction and the manual evaluation. It outlines the entity detection performance across individual entity types, summarises the detected properties, and discusses qualitative observations from the manual evaluation.

6.4.1 Entity Detection

The prototype was developed to identify entities belonging to the categories *person*, *organisation*, and *event*.

As described in Chapter 5, in an initial processing step, entities were extracted using Named Entity Recognition (NER) implemented with spaCy and Microsoft Azure. The outputs of both NER systems were subsequently combined, such that any entity detected by either spaCy or Azure was included in the final set of recognised entities. In a subsequent step, the identified entities served as the basis for extracting factual information related to each entity. Finally, the extracted results were cleaned by removing facts that contained empty entries or entries with fewer than two properties since they are not suitable for subsequent fact-checking tasks.

The following section presents the number of detected entities per category within the data set, as well as the number of entities that were not present in the fact output.

Detected entities

Figure 6.3 presents the number of entities detected in the data set for each of the three entities: *person*, *organisation*, and *event*. The results indicate that the highest number of detected entities was observed for *event* with a total of 116 entities, and *person* with a total of 115 entities. In contrast, the number of results for *organisation* is considerably lower, at 61 entities.

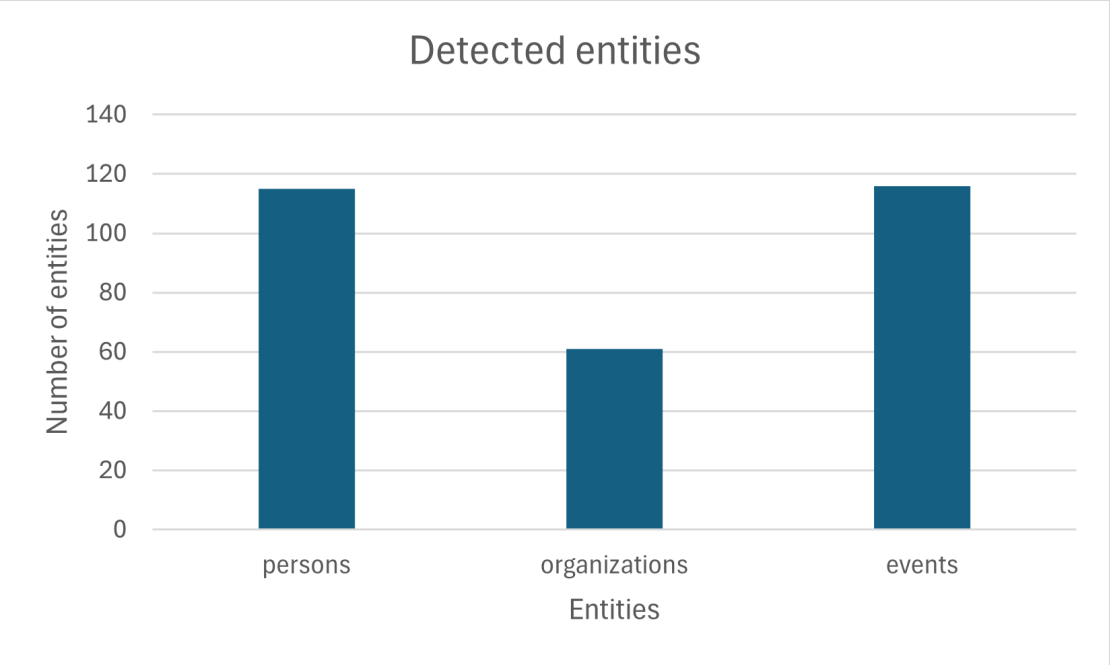


Figure 6.3: Detected entities

Undetected entities

Figure 6.4 shows the number of entities that were not detected by the extraction process. Therefore, the named entity recognition results based on the transcript summary were used to obtain comparison metrics. It was compared how many of the detected entities from the summary were brought to a fact result. As a result, 14 *person* entities, 4 *event* entities, and 29 *organisation* entities were missed. The relatively high number of undetected organisations suggests that this entity type posed the greatest challenge for the prototype. This matches the results of the detected entities, where the fewest were detected for the category *organisation*.

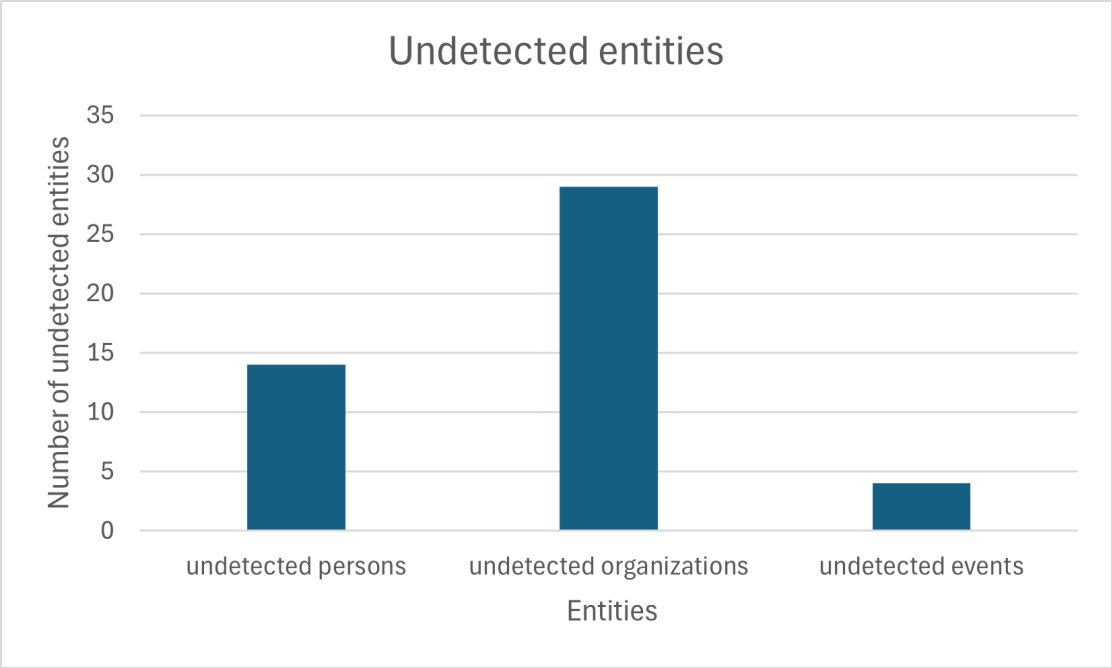


Figure 6.4: Undetected entities

Conclusion

The entity detection results show notable performance differences across the analysed entity types. The entities *event* and *person* were detected with nearly equal counts, 116 and 115, respectively. In comparison, the detection rate for *organisation* entities was substantially lower, with only 61 entities identified. This may suggest that fewer organisations were mentioned in the news video of the data set; however, the analysis of the undetected entities indicates a performance weakness for the entity type *organisation* compared to *person* and *event*. While relatively few *person* (10.85%) and *event* (3.33%) entities identified in the transcript summary were not structured into facts, approximately one third of the *organisation* entities (32.22%) were missed. These findings suggest that the current extraction process performs reliably for *person* and *event* entities, whereas the recognition of *organisation* entities remains a challenge. A possible explanation for this result is that *organisation* entities were detected by the prototype but were later removed when they were associated with fewer than two properties. This might occur if organisations were briefly mentioned in a news video, but the report did not provide sufficient detail about the organisation.

Overall, the results indicate that improvements to the entity extraction process should prioritise *organisation* entities to achieve more balanced, comprehensive extraction performance across all entity types.

6.4.2 Detected Properties

A central task of the automatic fact extraction was to summarise the available information about the entities *person*, *organisation*, and *event* into a structured fact representation. Therefore, the information available on these entities was stored by using properties (e.g., name, age, location). Furthermore, the prototype was designed to employ properties defined by *schema.org*, but was not strictly limited to them. This section presents the results of property detection for each entity.

Figure 6.5 illustrates the number of distinct properties extracted for each entity category in the data set. The results show that organisations exhibit the highest number of distinct properties (47), followed by persons with 35 properties, while events are associated with the fewest distinct properties (23).

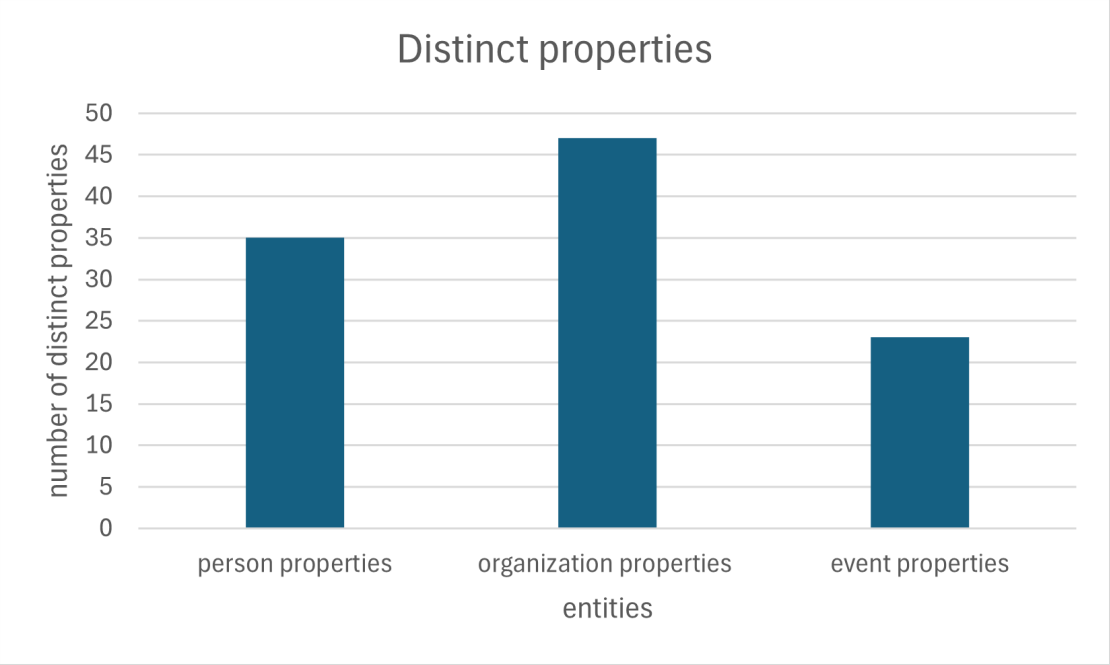


Figure 6.5: Distinct properties

Person

As outlined above, the *person* entity was the second-most frequently detected entity after *event*, with 115 occurrences, while 14 person instances were not detected. In total, 35 distinct properties were identified, and 379 property instances were used to describe the 115 identified persons, resulting in an average of 3.3 properties per person.

The distribution of properties is strongly imbalanced, with a small number accounting for most occurrences, while the majority appear only infrequently. The following section outlines the most frequently used properties and evaluates their conformance with

schema.org.

Conformance with schema.org Approximately two-thirds of the identified properties (24 out of 35) conform to the *schema.org* definition of the *person* entity, while one-third (11 properties) do not. Properties that conform to *schema.org* account for the majority of all property occurrences, whereas non-conforming properties appear less frequently, often only once. This indicates that the content of news videos about persons can largely be summarised using a standardised set of properties; however, capturing more specialised or rare information requires the use of customised properties.

The most frequently referenced properties conforming to *schema.org* are *name* (111), *jobTitle* (88), and *description* (36). Other notable properties include *affiliation* (21), *nationality* (20), and *occupation* (15).

Non-conforming properties that occurred only once are, for example, *medicalCondition* (1), *disability* (1), and *causeOfDeath* (1).

Conforming Properties The following properties conform to the *schema.org* definition for the *person* entity:

- *name* (111), *jobTitle* (88), *description* (36)
- *affiliation* (21), *nationality* (20), *occupation* (15)
- *memberOf* (10), *worksFor* (10), *spouse* (9)
- *hasOccupation* (6), *birthPlace* (5), *gender* (5)
- *honorificPrefix* (4), *deathDate* (3), *award* (3), *address* (3)
- *knowsAbout* (2), *honorificSuffix* (2)
- *children* (1), *deathPlace* (1), *alternateName* (1), *homeLocation* (1),
- *url* (1), *interactionCount* (1)

Non-Conforming Properties The following properties do not conform to the *schema.org* definition for the *person* entity:

- *age* (7), *location* (3), *employer* (2)
- *medicalCondition* (1), *disability* (1), *causeOfDeath* (1)
- *event* (1), *startDate* (1), *eventAttending* (1)
- *honorificTitle* (1), *hasRole* (1)

Property Performance Figure 6.6 illustrates the ratio of correctly (*true*) and incorrectly (*false*) attributed values for the detected properties of *person* entities. For nearly all properties, correctly attributed values vastly outnumber incorrect ones, indicating generally accurate property assignment. Overall, property assignment for *person* entities was highly successful. Of the 379 property assignments, 369 (approximately 97.4%) properties were correctly attributed, while only 10 (about 2.6%) were incorrect. Incorrect assignments occurred only for a small subset of distinct properties, which are listed in Table 6.2.

property	**true**	**false**
name	107	4
jobTitle	86	2
nationality	19	1
spouse	8	1
alternateName	0	1
url	0	1

Table 6.2: False properties: person

Organization

For the *organisation* entity, 61 instances were detected, and 29 were missed, highlighting the difficulty of *organisation* entity detection noted above. Across these entities, 47 distinct properties were identified, and 198 property instances were used to describe the detected organisations, resulting in an average of approximately 3.2 properties per organisation.

As with the *person* entity, the distribution of properties is similarly imbalanced, with a small number of properties accounting for most occurrences, while the majority of properties appear only infrequently. This tendency is even more evident for organisations than it was for persons. The following section outlines the most frequently used properties and evaluates their conformance with *schema.org*.

Conformance with schema.org The conformance with *schema.org* is less present than it was the case for persons. Approximately half of the identified properties (24 out of 47) conform to the *schema.org* definition of the *organisation* entity, while the other half (23 properties) do not. However, properties that conform to *schema.org* still account for the majority of all property occurrences, whereas non-conforming properties appear less frequently, in all cases only once or twice.

The most frequently referenced properties conforming to *schema.org* are *name* (58), *location* (20), and *description* (18). Other conforming properties referenced a few times include *founder* (9), *foundingLocation* (9), and *areaServed* (8).

Most of the non-conforming properties occurred only once, such as *numberOfParticipants* (1), *organizationType* (1), *ownershipStatus* (1), and *telephone* (1).

Compared to the properties of the *person* entity, the properties for *organisation* are more diverse. Most properties occur only a few times or even once. This suggests that the content of news videos about organisations is rather specialised and can only be captured using more customised properties than needed for news about persons.

Conforming Properties The following properties conform to the *schema.org* definition for the *organization* entity:

- *name* (58), *location* (20), *description* (18)
- *founder* (9), *foundingLocation* (9), *areaServed* (8)
- *employee* (7), *member* (5), *address* (5), *memberOf* (5)
- *numberOfEmployees* (4), *event* (3), *dissolutionDate* (3), *award* (3)
- *foundingDate* (3), *parentOrganization* (3), *department* (2)
- *email* (1), *naics* (1), *alternateName* (1)
- *founders* (1), *object* (1), *legalName* (1), *owns* (1)

Non-Conforming Properties The following properties do not conform to the *schema.org* definition for the *organization* entity:

- *leader* (2), *owner* (2), *legalAction* (2)
- *numberOfParticipants* (1), *financialStatus* (1), *logo* (1), *contract* (1), *brand* (1)
- *status* (1), *organizationType* (1), *ownershipStatus* (1), *controversy* (1)
- *leadership* (1), *courtCase* (1), *purpose* (1), *profit* (1), *investigation* (1)
- *hasCourtCase* (1), *statement* (1), *involvedPersons* (1)
- *telephone* (1), *keyPeople* (1), *knowsAbout* (1)

Property Performance Figure 6.7 illustrates the ratio of correctly (*true*) and incorrectly (*false*) attributed values for the detected properties of *organisation* entities. While property assignment for *person* entities was mostly accurate, *organisation* entities exhibit more errors. Of the 198 cases, 163 properties (approximately 82.3%) were assigned correctly, while 35 properties (about 17.7%) were attributed incorrectly. The incorrect assignments are listed in Table 6.3.

property	true	false
name	50	8
founder	4	5
foundingLocation	5	4
numberOfEmployees	0	4
areaServed	6	2
dissolutionDate	1	2
location	19	1
employee	6	1
memberOf	4	1
address	4	1
foundingDate	2	1
telephone	0	1
legalName	0	1
naics	0	1
email	0	1
logo	0	1

Table 6.3: False properties: organisation

Event

For the *event* entity, 116 instances were detected, while 4 *event* instances were missed. Across these entities, 23 distinct properties were identified, and 411 property instances were used to describe the detected events, resulting in an average of approximately 3.5 properties per event.

As with the *person* and *organisation* entities, the distribution of *event* properties is strongly imbalanced, with a small number of properties accounting for most occurrences, while the majority of properties appear only infrequently. The following section outlines the most frequently used properties and evaluates their conformance with *schema.org*.

Conformance with schema.org All properties that conform to *schema.org* account for the majority of all property occurrences, whereas non-conforming properties appear much less frequently. Properties of the *event* entity conform to *schema.org* more frequently than those of the other two entity types. 18 out of the 23 detected properties conform to the *schema.org* definition for the *event* entity, while only the remaining 5 do not. This indicates that news video content about events can largely be captured using a standardised set of properties, with only a small number requiring customised or specialised attributes.

The most frequently referenced properties conforming to *schema.org* are *name* (110), *location* (82), *result* (64), *date* (56), and *description* (39). Less frequent properties conforming to *schema.org* include *impact* (6), *about* (5), *eventStatus* (5), *organizer* (4), and *participant* (4).

Non-conforming properties, appearing only once, include *nameOfHoliday*, *allegations*,

and *lawsuit*.

Conforming Properties The following properties conform to the *schema.org* definition for the *event* entity:

- *name* (113), *location* (85), *result* (65), *date* (56), *description* (41),
- *startDate* (7), *eventStatus* (5), *about* (5), *participant* (4), *organizer* (4)
- *agent* (3), *endDate* (3), *cause* (3), *eventType* (2), *winner* (2)
- *mentions* (1), *status* (1), *participants* (1)

Non-Conforming Properties The following properties do not conform to the *schema.org* definition for the *event* entity:

- *impact* (6), *nameOfHoliday* (1), *allegations* (1), *Cancer* (1), *lawsuit* (1)

Property Performance Figure 6.8 illustrates the ratio of correctly (*true*) and incorrectly (*false*) attributed values for the detected properties of *event* entities. Property assignment for events was largely accurate, although some properties show higher error rates. Of the 411 property assignments, 371 (approximately 90.3%) were correctly attributed, while 40 (about 9.7%) were assigned incorrectly. Compared to *person* and *organisation* entities, the *event* entity shows the highest assignment accuracy, although temporal properties such as *date* (27 incorrect), *startDate* (3 incorrect), and *endDate* (1 incorrect) present the most challenges. All incorrect assignments are listed in Table 6.4.

property	**true**	**false**
date	29	27
location	82	3
name	110	3
startDate	4	3
description	39	2
endDate	2	1
result	64	1

Table 6.4: False properties: event

6.4.3 Qualitative Observations

This section presents qualitative insights into the performance of the prototype, highlighting recurring patterns and shortcomings in the extraction of facts from news videos. The observations focus on the accuracy, specificity, and meaningfulness of extracted property values, with particular attention to issues such as misspellings of names, unspecific or

overly broad descriptions, temporal ambiguities, and partially incorrect values. Looking at representative examples, the qualitative analysis shows the main limitations of the prototype and suggests ways of improvement.

Frequent Misspellings of Names

A common issue was the frequent misspelling of the *name* property. This often resulted in the same individual being represented multiple times with slightly different spellings. Such errors were particularly common when names were mentioned verbally in the news video, rather than appearing as OCR text, and subsequently transcribed in multiple versions. Representative examples are presented in Table 6.5.

Incorrect name	**Correct name**
Michelle Georgina Moan (Baroness_Mone_linked_person_4)	Michelle Georgina Mone (Baroness_Mone_linked_person_7)
Sumi Somerskander (Nancy_Pelosi_announc_person_6)	Sumi Somaskanda (Nancy_Pelosi_announc_person_8)
Khuzi Hari (Peru_to_declare_stat_person_3)	José Jerí (Peru_to_declare_stat_person_4)
Senai Takaichi (Sanae_Takaichi_to_be_person_5)	Sanae Takaichi (Sanae_Takaichi_to_be_person_6)
Caroline Davis (Taliban_imposes_inte_person_1)	Caroline Davies (Taliban_imposes_inte_person_2)
PPE Medro Med Pro company (Baroness_Mone_linked_organization_2)	PPE Medpro (Baroness_Mone_linked_organization_6)

Table 6.5: Fact example: misspelling

Extraction of Meaningless Property Values

Another recurring weakness concerned the quality of extracted property values. In several cases, property values lacked meaningful content. For example, as shown in Table 6.6, the *foundingLocation* property was filled with the placeholder text *"Not answered in the article"*, which does not contribute to information gain. Table 6.7 presents another instance in which the *url* property contains the value *"Her website is mentioned in the article"*. Although this extraction is consistent with the news video content, it does not meet the requirements of a valid *url* value.

To address this issue, a cleaning algorithm was introduced during implementation to filter out meaningless property values; however, in several cases, it did not perform as intended.

id	**Hundreds_of_National_organization_6**
name	BB C
name - evaluation	0
foundingLocation	Not answered in the article
foundingLocation - evaluation	0

Table 6.6: Fact example: meaningless "foundingLocation"

Unspecific Temporal Properties

Temporal properties posed a particular challenge for the prototype. Properties such as *date*, *startDate*, *endDate*, and *dissolutionDate* performed poorly across multiple examples. Rather than extracting concrete and comparable timestamps, the prototype frequently produced vague temporal expressions. Some examples include the *date* value *"Yesterday morning local time"* (Table 6.8), the *endDate* value *"Verdict given recently"* (Table 6.9), the *date* value *"Friday"* (Table 6.10), the *date* value *"recent"* (Table 6.11), and the *dissolutionDate* value *"Just in the last hour"* (Table 6.12). Although these expressions reflect the wording used in the news videos, they are unsuitable for structured comparison and were therefore evaluated as incorrect.

This limitation stems from common news reporting practices, where precise timestamps are often omitted because the temporal proximity of the report to the event is implicitly assumed. Consequently, the prototype struggled to infer precise temporal properties from the available input. The implementation quality could be improved by incorporating an algorithm that resolves implicit temporal expressions into concrete timestamps.

Overly Broad or Unspecific Descriptions

Another weakness concerns the extraction of overly broad or unspecific values for the *description* property. These descriptions frequently combine multiple, distinct aspects into a single property value. One example, shown in Table 6.13, contains the description *"Had foot amputated after accident, Paralympic athlete, campaigner"* that combines the medical history and the profession of the *person* entity. A similar issue can be observed in Table 6.14, where the *description* property *"Survived no confidence votes, postponed pension reform"* bundles two political events into one value. Although these descriptions align with the content of the news video, their lack of granularity and focus make them unsuitable for future fact-checking tasks.

Table 6.15 presents another example of an overly broad *description* value. The extracted text, *"Peaceful protests following arrival of federal immigration officials"*, accurately reflects the content of the news video; however, it is too verbose and lacks sufficient focus on the key aspect, making it unsuitable for reliable fact-checking.

To address this issue, it is necessary to clearly define the content requirements of the *description* property.

id	**Jane_Goodall_chimpan_person_1**
name	Jane Goodall
name - evaluation	1
jobTitle	Conservationist, ethologist, chimpanzee expert
jobTitle - evaluation	1
deathDate	Died at age 91, recently announced
deathDate - evaluation	1
description	World-renowned chimpanzee and nature expert
description - evaluation	1
affiliation	Jane Goodall Institute
affiliation - evaluation	1
nationality	British, grew up in London
nationality - evaluation	1
worksFor	Jane Goodall Institute
worksFor - evaluation	1
url	Her website is mentioned in article
url - evaluation	0

Table 6.7: Fact example: meaningless "url"

Partially Incorrect Property Values

Some extracted properties are only partially correct and, therefore, unsuitable for reliable fact comparison.

The example shown in Table 6.16 illustrates a property that contains information that is only partially accurate. The *description* property includes the statement *"Odinga said polls were rigged; elections in two years."* This extraction is considered partially correct because the news video confirms that Odinga stated the polls were rigged. However, the second part of the statement, referring to elections occurring in two years, is incorrect, as this information is not mentioned in the source material. As a result, although parts of

id	**France_reels_after_L_event_1**
name	Raid/Robbery at the Louvre
name - evaluation	1
location	Louvre Museum, Paris
location - evaluation	1
date	Yesterday morning local time
date - evaluation	0

Table 6.8: Fact example: unspecific "date"

id	**Brazil_s_former_Pres_event_1**
name	trial
name - evaluation	1
about	Trial of Bolsonaro for coup attempt
about - evaluation	1
result	Bolsonaro found guilty
result - evaluation	1
location	Brasilia
location - evaluation	1
endDate	Verdict given recently
endDate - evaluation	0

Table 6.9: Fact example: unspecific "endDate"

id	**No_immediate_plans_f_event_1**
name	Meeting between Trump and Zelensky
name - evaluation	1
location	Washington
location - evaluation	1
date	Friday
date - evaluation	0

Table 6.10: Fact example: "Friday" as unspecific "date"

the extracted property are accurate, the presence of incorrect information renders the property unreliable for further fact comparison.

In another example shown in Table 6.17 Donald Trump's *jobTitle* is extracted as *"Former President of the United States"*. While this property is generally true in the broader temporal context that he was also the president in a former election period, it is inaccurate with respect to the time frame of the analysed news video. At the time of the video's release, Donald Trump was serving as the current President of the United States, and the video content does not provide any indication that he was a former president. Consequently, the extracted *jobTitle* property is considered partially correct but inconsistent with the source material, rendering it unsuitable for further fact-based evaluation.

Over-representation of Broadcasting Company

The broadcasting company BBC was frequently interpreted as an *organisation* of central relevance to the news content. This issue is demonstrated in Table 6.18 and Table 6.19 where *BBC News* is extracted as *organization* entity with a generic description like *"Global news organization reporting on [...]"*. While the description is technically correct, it does not add meaningful contextual information about the *organization* entity itself. These

id	**French_ex_president__event_1**
name	Nicolas Sarkozy's sentencing and imprisonment
name - evaluation	1
location	Sante prison, Paris
location - evaluation	1
date	recent
date - evaluation	0

Table 6.11: Fact example: "recent" as unspecific "date"

id	**Madagascar_President_organization_1**
name	National Assembly
name - evaluation	1
dissolutionDate	Just in the last hour
dissolutionDate - evaluation	0
location	Madagascar
location - evaluation	1

Table 6.12: Fact example: unspecific "dissolutionDate"

extractions inflate the number of entities without contributing to a deeper understanding of the reported facts.

6.5 Discussion

This chapter presented the results of the evaluation of the prototype for automatic fact extraction. The analysed data set consisted of 31 English-language news videos published by the BBC.

The evaluation was conducted by first using the prototype to automatically extract facts and subsequently manually validating the results. The prototype focused on identifying

id	**Paralympic_athlete_c_person_2**
name	Stef Reid
name - evaluation	1
occupation	Former Para athletic world champion
occupation - evaluation	1
description	Had foot amputated after accident, Paralympic athlete, campaigner
description - evaluation	1

Table 6.13: Fact example: unspecific description

id	**French_PM_S_bastien__person_5**
name	Sebastian de Corneau
name - evaluation	1
jobTitle	Prime Minister of France
jobTitle - evaluation	1
affiliation	President Macron
affiliation - evaluation	1
employer	Government of France
employer - evaluation	1
hasOccupation	Prime Minister
hasOccupation - evaluation	1
description	Survived no confidence votes, postponed pension reform.
description - evaluation	1

Table 6.14: Fact example: unspecific "description" value

id	**Hundreds_of_National_event_2**
name	protests
name - evaluation	1
location	Chicago, Illinois, United States
location - evaluation	1
description	Peaceful protests following arrival of federal immigration officials.
description - evaluation	1

Table 6.15: Fact example: unspecific property value

person, *organisation*, and *event* entities, which were then structured into facts using appropriate properties.

The results indicate that *person* and *event* entities were detected reliably, whereas *organisation* entities were more frequently missed. The number of missed entities was determined by comparing the entities identified in the video summary with those represented in the fact results. While only a small proportion of *person* (10.85%) and *event* (3.33%) entities identified in the transcript summary were not structured into facts, nearly one third of the *organisation* entities (32.22%) were not covered by the final result. This may be because some organisations were not detected at all, or because they were detected but later removed when they were linked to fewer than two properties. One possible explanation is that organisations were mentioned only briefly in the news video and were not accompanied by enough detail to support fact extraction. Additionally, because the final fact structuring performed by artificial intelligence cannot be fully controlled, it may produce indeterminate results.

Property extraction achieved high correctness for *person* (approximately 97.4%) and

id	**Former_Kenyan_Prime__event_2**
eventType	Elections
eventType - evaluation	1
description	Odinga said polls were rigged; elections in two years.
description - evaluation	0
location	Kenya
location - evaluation	1

Table 6.16: Fact example: partially incorrect property value for "description"

id	**Nancy_Pelosi_announc_person_3**
name	Donald Trump
name - evaluation	1
jobTitle	Former President of the United States
jobTitle - evaluation	0

Table 6.17: Fact example: partially incorrect property value for "jobTitle"

event (approximately 90.3%) entities. A major source of errors was temporal properties such as *date*, *startDate*, and *endDate*, which frequently contained unspecific values. Most extracted properties conformed to *schema.org* standards; however, for certain types of specialised information present in the news videos, the use of non-conforming properties was necessary for both entity types.

In contrast, *organisation* entities proved more challenging, achieving a correctness rate of approximately 82.3%. About half of the identified properties did not conform to *schema.org*, indicating that information about organisations in the analysed news videos often required more specialised, non-standard properties.

Qualitative issues observed during the evaluation included misspellings of names, vague temporal expressions, overly broad descriptions, partially incorrect property values, and the over-representation of the broadcasting company.

Future improvements should focus on obtaining more precise temporal property values, normalising entity names to reduce duplication, validating property values to remove

id	**Former_Kenyan_Prime__organization_1**
name	BBC News
name - evaluation	1
description	Global news organization reporting Raila Odinga's death.
description - evaluation	1

Table 6.18: Fact example: broadcasting company

id	**Peru_to_declare_stat_organization_4**
name	BBC News
name - evaluation	1
description	Global news organization reporting on Peru emergency
description - evaluation	1

Table 6.19: Fact example: over-represented broadcasting company

meaningless entries, and improving the identification of *organisation* entities. Implementing these improvements is expected to increase the reliability of generated fact reports, enabling better fact extraction suitable for more accurate fact comparison of news video content.

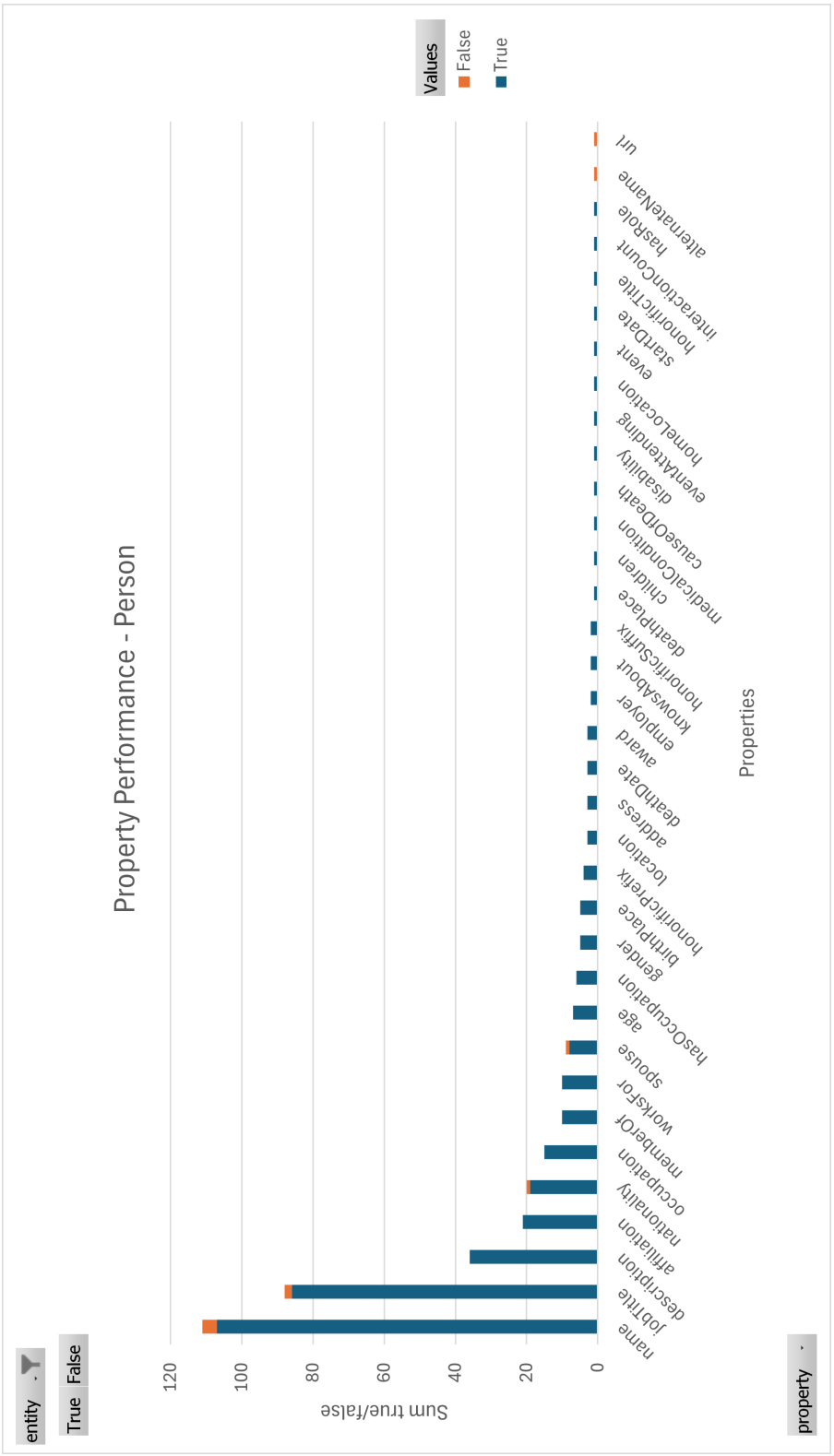


Figure 6.6: Property performance: person

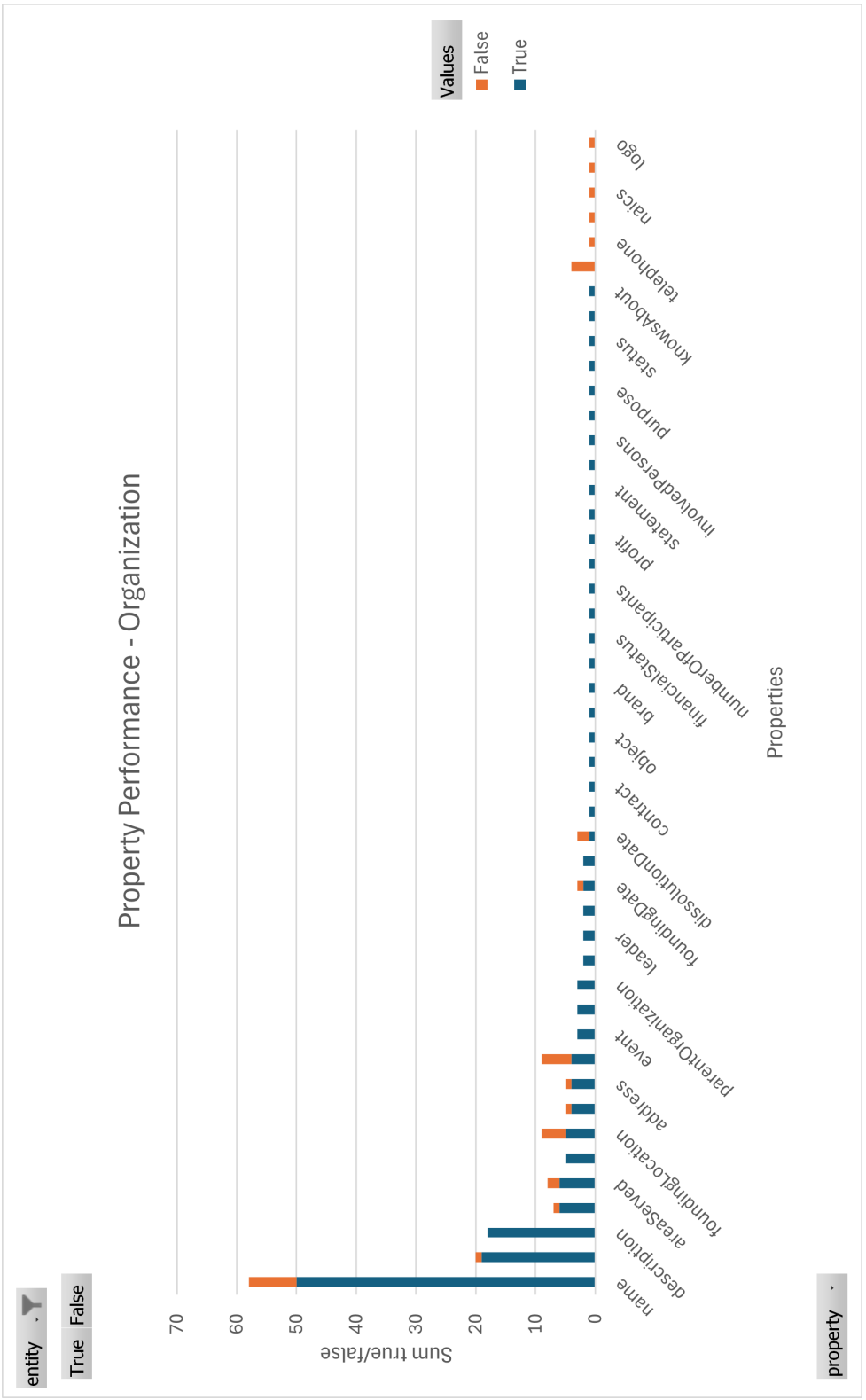


Figure 6.7: Property performance: organisation

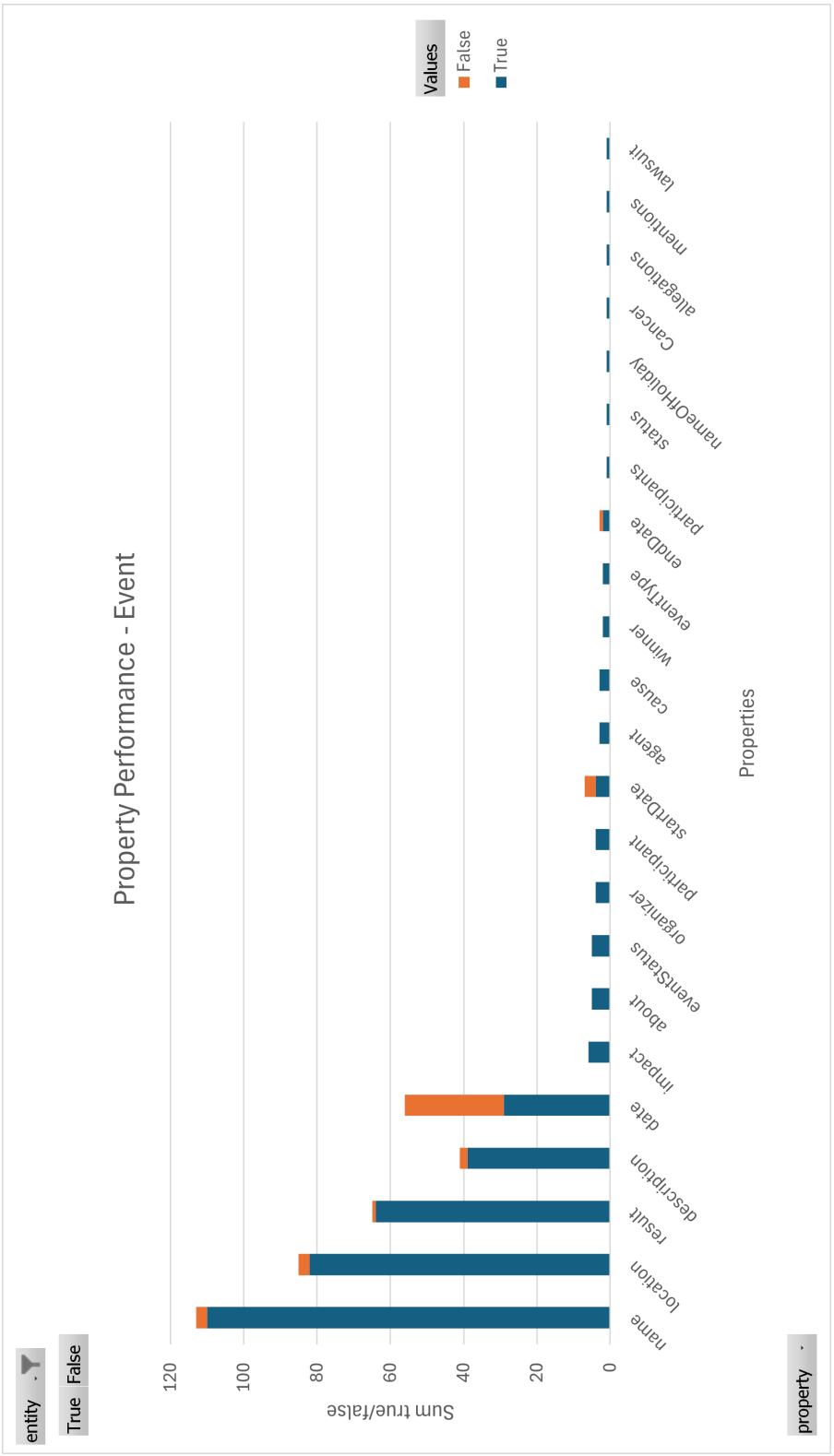


Figure 6.8: Property performance: event

7 Conclusion

This master's thesis addressed the challenge of extracting and structuring information from news videos. It focused on identifying *person*, *organisation*, and *event* entities, along with their properties. The main goal was to build a system that automatically transforms unstructured video into structured facts suitable for fact-checking applications.

The fact extraction prototype integrates speech transcription, text recognition, and entity-property mapping. It uses these parts to capture relevant information from news videos. The system organises this information using a structured schema suitable for fact-checking.

The quantitative evaluation showed high correctness for *person* and *event* entities. Coverage was lower for *organisation* entities. Extracting temporal, descriptive, and relational properties showed both strengths and weaknesses in automated extraction.

The qualitative evaluation identified several common problems, including misspelt names, unclear or overly general dates and descriptions, partially incorrect property values, meaningless results, and excessive references to broadcasting companies. These findings show where the fact extraction system can be improved in the future.

Future improvement recommendations include standardising entity names and improving date and time recognition. They also suggest checking the accuracy of the extracted values and handling complex or combined descriptions more precisely. These steps can improve the accuracy and reliability of fact extraction.

7.1 Research Question Findings

The research questions posed in this thesis have been addressed as follows:

1. Structuring unstructured data: The prototype shows that multimodal information from news videos can be extracted and represented in a structured form. Some entities and properties were omitted or only partly correct, but the approach is scalable for capturing core facts.

2. Organising key facts: Grouping extracted information into entities and properties reduces redundancy and highlights the main news content. However, broad or compound descriptions suggest a need to refine property definitions and summarisation strategies.

3. Prototype implementation for fact extraction and verification: The prototype shows that automated fact extraction from news videos is possible. It provides a base for adding more extraction mechanisms in future work.

7.2 Limitations

Despite the above-mentioned achievements, several limitations were observed:

- Temporal property resolution: Many extracted timestamps were vague, reflecting the common practice in news reporting of omitting precise time information.
- Organisation entity extraction: Coverage for *organisation* entities was lower than for *person* and *event entities*. This may be due to brief mentions and insufficient contextual information in the videos, or due to non-controllable fact structuring performed by artificial intelligence.
- Property correctness: Some properties were partially incorrect, overly broad, or meaningless, making them unsuitable for future fact comparison tasks.
- Dependency on input quality: The accuracy of speech transcription and text recognition directly impacts entity and property extraction, highlighting the sensitivity of the system to input quality.

7.3 Future Work

Building on the findings of this thesis, several directions for future research are proposed.

One important area is improving temporal values, where algorithms could be implemented to resolve vague temporal expressions into concrete timestamps, thereby enabling more accurate fact comparison.

Another direction is normalising entity names. This would mean correcting misspellings and merging duplicate entities. Property validation and cleaning could also be advanced by filtering out meaningless or partly incorrect values.

Future work should also focus on improving extraction for *organisation* entities by further investigating the causes of the entity type's poor performance, which could be due to only short mentions of organisations in the video material or to unpredictable performance of the fact-structuring by artificial intelligence, leading to the removal of unsuitable facts by the prototype.

Finally, the prototype could be extended to support additional entity types beyond *person*, *organisation*, and *event*.

Bibliography

[1] Newman, Arguedas, Robertson, Nielsen, Fletchers, "Digital News Report 2025," Reuters Institute for the Study of Journalism, 2025. [Online]. Available: https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2025.

[2] F. Rayar, "Fact-Checked Claim Detection in Videos Using a Multimodal Approach," in *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Rome, Italy: SCITEPRESS - Science and Technology Publications, 2024, pp. 614–620, ISBN: 978-989-758-679-8. DOI: 10.5220/0012545500003660. [Online]. Available: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0012545500003660 (visited on 03/03/2026).

[3] D. Giomelakis, O. Papadopoulou, S. Papadopoulos and A. Veglis, "Verification of News Video Content: Findings from a Study of Journalism Students," *Journalism Practice*, vol. 17, no. 5, pp. 1068–1097, 28th May 2023, ISSN: 1751-2786, 1751-2794. DOI: 10.1080/17512786.2021.1965905. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/17512786.2021.1965905 (visited on 11/09/2024).

[4] R. Kozik, A. Pawlicka, M. Pawlicki, M. Choraś, W. Mazurczyk and K. Cabaj, "A Meta-Analysis of State-of-the-Art Automated Fake News Detection Methods," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 5219–5229, Aug. 2024, ISSN: 2329-924X, 2373-7476. DOI: 10.1109/TCSS.2023.3296627. [Online]. Available: https://ieeexplore.ieee.org/document/10196320/ (visited on 01/11/2024).

[5] M. Akhtar, M. Schlichtkrull, Z. Guo, O. Cocarascu, E. Simperl and A. Vlachos, "Multimodal Automated Fact-Checking: A Survey," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore: Association for Computational Linguistics, 2023, pp. 5430–5448. DOI: 10.18653/v1/2023.findings-emnlp.361. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.361 (visited on 03/03/2026).

[6] S. Jiang and C. Wilson, "Structurizing Misinformation Stories via Rationalizing Fact-Checks," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, 2021, pp. 617–631. DOI: 10.18653/v1/2021.acl-long.51. [Online]. Available: https://aclanthology.org/2021.acl-long.51 (visited on 07/11/2024).

[7] E. Aïmeur, S. Amri and G. Brassard, "Fake news, disinformation and misinformation in social media: a review," *Social Network Analysis and Mining*, vol. 13, no. 1, p. 30, 9th Feb. 2023, ISSN: 1869-5469. DOI: 10.1007/s13278-023-01028-5. [Online]. Available: https://link.springer.com/10.1007/s13278-023-01028-5 (visited on 11/11/2024).

[8] A. Sängerlaub, "Im Zeitalter von Fake News. Warum sich der (Nachrichen-)Journalismus neu erfinden muss," in *Fake news, Framing, Fact-Checking: Nachrichten im digitalen Zeitalter. Ein Handbuch*, T. Köhler, Ed., Bielefeld: Transcript, 2020.

[9] J. Stern, "Fact-Checking und Verifikation. Neue Formen des Faktenprüfens im Nachrichtenjournalismus," in *Fake news, Framing, Fact-Checking: Nachrichten im digitalen Zeitalter. Ein Handbuch*, T. Köhler, Ed., Bielefeld: Transcript, 2020.

[10] Z. Guo, M. Schlichtkrull and A. Vlachos, "A Survey on Automated Fact-Checking," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 178–206, 9th Feb. 2022, ISSN: 2307-387X. DOI: 10.1162/tacl_a_00454. [Online]. Available: https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00454/109469/A-Survey-on-Automated-Fact-Checking (visited on 26/10/2024).

[11] X. Zhou and R. Zafarani, "A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–40, 30th Sep. 2021, ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3395046. [Online]. Available: https://dl.acm.org/doi/10.1145/3395046 (visited on 01/11/2024).

[12] "Politifact," Politifact. (), [Online]. Available: https://www.politifact.com/.

[13] "Snopes," Snopes. (), [Online]. Available: https://www.snopes.com/.

[14] "Amazon Mechanical Turk," Amazon Mechanical Turk. (), [Online]. Available: https://www.mturk.com/.

[15] B. Marcus, "Vom Nachrichtentanker zum Schnellboot. Die Tagesschau zwischen öffentlich-rechtlichem Auftrag und Marktanforderung," in *Fake News, Framing, Fact-Checking: Nachrichten im digitalen Zeitalter : Ein Handbuch*, ser. Digitale Gesellschaft ; 30, Bielefeld : transcript-Verlag, 2020, ISBN: 978-3-8394-5025-3.

[16] H. Burger and M. Luginbühl, *Mediensprache: Eine Einführung in Sprache und Kommunikationsformen der Massenmedien*. De Gruyter, 29th Jan. 2014, ISBN: 978-3-11-037242-7. DOI: 10.1515/9783110285925. [Online]. Available: https://www.degruyter.com/view/title/124154 (visited on 20/10/2024).

[17] A. Kalogeropoulos and R. K. Nielsen, "Investing in Online Video News," *Journalism Studies*, vol. 19, no. 15, pp. 2207–2224, 18th Nov. 2018, ISSN: 1461-670X. DOI: 10.1080/1461670X.2017.1331709. [Online]. Available: https://doi.org/10.1080/1461670X.2017.1331709.

[18] A. Kalogeropoulos, "Online News Video Consumption," *Digital Journalism*, vol. 6, no. 5, pp. 651–665, 28th May 2018, ISSN: 2167-0811. DOI: 10.1080/21670811.2017.1320197. [Online]. Available: https://doi.org/10.1080/21670811.2017.1320197.

[19] R. Fletcher *et al.*, "The link between changing news use and trust: longitudinal analysis of 46 countries," *Journal of Communication*, vol. 75, no. 1, pp. 1–15, 1st Feb. 2025, ISSN: 0021-9916, 1460-2466. DOI: 10.1093/joc/jqae044. [Online]. Available: https://academic.oup.com/joc/article/75/1/1/7907139 (visited on 10/02/2026).

[20] J. Wang, "A Study of The OCR Development History and Directions of Development," *Highlights in Science, Engineering and Technology*, vol. 72, pp. 409–415, 15th Dec. 2023, ISSN: 2791-0210. DOI: 10.54097/bm665j77. [Online]. Available: https://drpress.org/ojs/index.php/HSET/article/view/13285 (visited on 22/11/2024).

[21] R. John. "What is Optical Character Recognition (OCR)?" Optical Character Recognition. (15th Nov. 2024), [Online]. Available: https://www.docsumo.com/blogs/ocr/definition (visited on 02/12/2024).

[22] Microsoft Azure. "OCR - Optical Character Recognition." (21st Nov. 2025), [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/overview-ocr (visited on 14/02/2026).

[23] Microsoft Azure. "OCR for images (version 4.0)." (18th Nov. 2025), [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/concept-ocr (visited on 14/02/2026).

[24] L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," *IEEE Access*, vol. 8, pp. 139 110–139 120, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3011028. [Online]. Available: https://ieeexplore.ieee.org/document/9145558/ (visited on 26/12/2024).

[25] H. L. Gururaj, B. C. Soundarya, S. Priya, J. Shreyas and F. Flammini, "A Comprehensive Review of Face Recognition Techniques, Trends, and Challenges," *IEEE Access*, vol. 12, pp. 107 903–107 926, 2024, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3424933. [Online]. Available: https://ieeexplore.ieee.org/document/10589377/ (visited on 26/12/2024).

[26] Microsoft Azure. "What is the Azure AI Face service?" (28th Aug. 2024), [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/overview-identity (visited on 26/12/2024).

[27] "Azure AI Video Indexer overview," Learn Azure. (18th Nov. 2025), [Online]. Available: https://learn.microsoft.com/en-us/azure/azure-video-indexer/video-indexer-overview (visited on 14/02/2026).

[28] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach* (Signals and Communication Technology). London: Springer, 2015, 321 pp., ISBN: 978-1-4471-5778-6 978-1-4471-5779-3. DOI: 10.1007/978-1-4471-5779-3.

[29] C. Arriaga, A. Pozo, J. Conde and A. Alonso, *Evaluation of real-time transcriptions using end-to-end ASR models*, 2024. DOI: 10.48550/ARXIV.2409.05674. [Online]. Available: https://arxiv.org/abs/2409.05674 (visited on 02/01/2025).

[30] S. Alharbi *et al.*, "Automatic Speech Recognition: Systematic Literature Review," *IEEE Access*, vol. 9, pp. 131 858–131 876, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3112535. [Online]. Available: https://ieeexplore.ieee.org/document/9536732/ (visited on 02/01/2025).

[31] Microsoft Azure. "Speech to text," What is speech to text? (12th Feb. 2026), [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/speech-service/speech-to-text (visited on 14/02/2026).

[32] R. Sharma, D. Govind, J. Mishra, A. K. Dubey, K. T. Deepak and S. R. M. Prasanna, "Milestones in speaker recognition," *Artificial Intelligence Review*, vol. 57, no. 3, p. 58, 15th Feb. 2024, ISSN: 1573-7462. DOI: 10.1007/s10462-023-10688-w. [Online]. Available: https://link.springer.com/10.1007/s10462-023-10688-w (visited on 11/01/2025).

[33] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, Aug. 2021, ISSN: 08936080. DOI: 10.1016/j.neunet.2021.03.004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0893608021000848 (visited on 11/01/2025).

[34] Microsoft Azure. "Speaker Recognition," What is speaker recognition? (24th Sep. 2024), [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/speech-service/speaker-recognition-overview (visited on 19/01/2025).

[35] *Azure AI video indexer*. [Online]. Available: https://azure.microsoft.com/en-us/products/ai-video-indexer.

[36] Z. Hu, W. Hou and X. Liu, "Deep learning for named entity recognition: a survey," *Neural Computing and Applications*, vol. 36, no. 16, pp. 8995–9022, Jun. 2024, ISSN: 0941-0643, 1433-3058. DOI: 10.1007/s00521-024-09646-6. [Online]. Available: https://link.springer.com/10.1007/s00521-024-09646-6 (visited on 25/01/2025).

[37] B. Jehangir, S. Radhakrishnan and R. Agarwal, "A survey on Named Entity Recognition — datasets, tools, and methodologies," *Natural Language Processing Journal*, vol. 3, p. 100 017, Jun. 2023, ISSN: 29497191. DOI: 10.1016/j.nlp.2023.100017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2949719123000146 (visited on 25/01/2025).

[38] *EntityRecognizer*. [Online]. Available: https://spacy.io/api/entityrecognizer.

[39] *Named Entity Recognition*. [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/language-service/named-entity-recognition/overview.

[40] Gonçalves, Luís. "Automatic Text Summarization with Machine Learning — An overview," Luisfredgs. (2020), [Online]. Available: https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25 (visited on 15/02/2026).

[41] Y. Zhang, H. Jin, D. Meng, J. Wang and J. Tan, *A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods*, 31st Oct. 2025. DOI: 10.48550/arXiv.2403.02901. arXiv: 2403.02901[cs]. [Online]. Available: http://arxiv.org/abs/2403.02901 (visited on 15/02/2026).

[42] *spaCy*. [Online]. Available: https://spacy.io/ (visited on 15/02/2026).

[43] *Azure Language Services Summarization*. [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/language-service/summarization/overview?tabs=text-summarization (visited on 15/02/2026).

[44] Y. Chen, H. Wang, K. Yu and R. Zhou, "Artificial Intelligence Methods in Natural Language Processing: A Comprehensive Review," *Highlights in Science, Engineering and Technology*, vol. 85, pp. 545–550, 13th Mar. 2024, ISSN: 2791-0210. DOI: 10.54097/vfwgas09. [Online]. Available: https://drpress.org/ojs/index.php/HSET/article/view/18422 (visited on 25/02/2026).

[45] S. K. Rongali, "Natural Language Processing (NLP) in Artificial Intelligence," *World Journal of Advanced Research and Reviews*, vol. 25, no. 1, pp. 2515–2519, 31st Jan. 2025, ISSN: 25819615. DOI: 10.30574/wjarr.2025.25.1.0277. [Online]. Available: https://journalwjarr.com/node/3251 (visited on 25/02/2026).

[46] *Azure OpenAI*. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/cognitive-services/openai-service/#overview.

[47] H. Choi and Y. Ko, "Effective fake news video detection using domain knowledge and multimodal data fusion on youtube," *Pattern Recognition Letters*, vol. 154, pp. 44–52, 2022, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2022.01.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865522000071.

[48] P. Qi, Y. Zhao, Y. Shen, W. Ji, J. Cao and T.-S. Chua, "Two Heads Are Better Than One: Improving Fake News Video Detection by Correlating with Neighbors," in *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada: Association for Computational Linguistics, 2023, pp. 11 947–11 959. DOI: 10.18653/v1/2023.findings-acl.756. [Online]. Available: https://aclanthology.org/2023.findings-acl.756 (visited on 15/02/2026).

[49] J. Thorne, A. Vlachos, C. Christodoulopoulos and A. Mittal, "FEVER: a Large-scale Dataset for Fact Extraction and VERification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 809–819. DOI: 10.18653/v1/N18-1074. [Online]. Available: http://aclweb.org/anthology/N18-1074 (visited on 03/03/2026).

[50] R. Aly *et al.*, "The Fact Extraction and VERification Over Unstructured and Structured information (FEVEROUS) Shared Task," in *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, Dominican Republic: Association for Computational Linguistics, 2021, pp. 1–13. DOI: 10.18653/v1/2021.fever-1.1. [Online]. Available: https://aclanthology.org/2021.fever-1.1 (visited on 03/03/2026).

[51] W. Y. Wang, ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, R. Barzilay and M.-Y. Kan, Eds., Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 422–426. DOI: 10.18653/v1/P17-2067. [Online]. Available: https://aclanthology.org/P17-2067/.

[52] S. Singh, *Natural Language Processing for Information Extraction*, 6th Jul. 2018. DOI: 10.48550/arXiv.1807.02383. arXiv: 1807.02383[cs]. [Online]. Available: http://arxiv.org/abs/1807.02383 (visited on 01/03/2026).

[53] Y. Yang, Z. Wu, Y. Yang, S. Lian, F. Guo and Z. Wang, "A Survey of Information Extraction Based on Deep Learning," *Applied Sciences*, vol. 12, no. 19, p. 9691, 27th Sep. 2022, ISSN: 2076-3417. DOI: 10.3390/app12199691. [Online]. Available: https://www.mdpi.com/2076-3417/12/19/9691 (visited on 01/03/2026).

[54] D. Y.-B. Wang, Z. Shen, S. S. Mishra, Z. Xu, Y. Teng and H. Ding, *SLOT: Structuring the Output of Large Language Models*, 6th May 2025. DOI: 10.48550/arXiv.2505.04016. arXiv: 2505.04016[cs]. [Online]. Available: http://arxiv.org/abs/2505.04016 (visited on 01/03/2026).

[55] M. R. DeVerna, H. Y. Yan, K.-C. Yang and F. Menczer, "Fact-checking information from large language models can decrease headline discernment," *Proceedings of the National Academy of Sciences*, vol. 121, no. 50, e2322823121, 10th Dec. 2024, ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.2322823121. [Online]. Available: https://pnas.org/doi/10.1073/pnas.2322823121 (visited on 01/03/2026).

[56] *Google Cloud Video Intelligence API*, 28th Feb. 2026. [Online]. Available: https://cloud.google.com/video-intelligence.

[57] *Amazon Rekognition Video*, 28th Feb. 2026. [Online]. Available: https://aws.amazon.com/de/rekognition/video-features/.

[58] *Clarifai*, 28th Feb. 2026. [Online]. Available: https://www.clarifai.com/.

[59] *Rev.ai Speech-To-Text*, 28th Feb. 2026. [Online]. Available: https://www.rev.ai/.

[60] *Schema.org*, 8th Dec. 2025. [Online]. Available: https://schema.org/.

[61] *Flask*. [Online]. Available: https://flask.palletsprojects.com/en/stable/.