BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION Applied Computational Intelligence

DISSERTATION THESIS

MindBugs Discovery XAI Search Engine

Supervisor
Prof.dr. Simona Motogna

Author
Chereș Ioana

2023

UNIVERSITATEA BABEŞ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
SPECIALIZAREA Inteligență computațională aplicată

LUCRARE DE DISERTAŢIE

MindBugs Discovery XAI Search Engine

Conducător științific
Prof.dr. Simona Motogna

Absolvent
Chereș Ioana

2023

ABSTRACT

MindBugs Discovery is a platform designed to combat misinformation by integrating state-of-the-art machine learning techniques with knowledge graph technology. The platform aggregates data from verified fact-checking organizations across Europe, providing a centralized repository of reliable information.

Utilizing a microservices architecture, the system is modular and highly scalable, enabling the addition of new data sources and analytical algorithms. A scalable Extract, Load, Transform (ELT) pipeline is designed and facilitated by dedicated web scrapers and a robust messaging queue, ensuring system reliability and adaptability.

The initiative presents a novel methodology for data collection. It uses community detection to enhance the precision of large language models like gpt-3.5 for specialized tasks. By leveraging community analysis, the methodology optimizes data segmentation subsequently feeding it into the model for fine-tuning. Empirical evaluation has indicated marked improvements in model performance when utilizing this refined dataset. The search inside the knowledge graph provides users with a community of related information for different queries. It uses a combination of Fast Random Projections and k-nearest Neighbours as an approach to optimize the time efficiency of the search.

Finally, the system presents a novel technique in data visualization. Instead of just presenting a catalog of results, MindBugs Discovery utilizes visualization methods to emphasize the connections between data points, providing a justification for the outcomes it produces. This facilitates a broader and more intuitive comprehension of the data, hence augmenting user involvement and fostering the development of analytical thinking skills.

Acknowledgments

I would like to extend my deepest gratitude to Ph.D. Simona Motogna, whose guidance, support, and expertise have been invaluable throughout the course of this research.

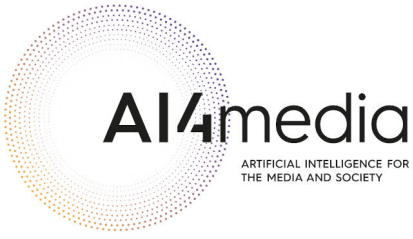
I am also profoundly grateful to Ph.D. Adrian Groza for fostering an environment of intellectual curiosity.

The line of projects in my work directed towards the XAI area started with "The Profile unleashing your deepfake self" which evoluated into the "Transparency" project. The whole idea of finding insights about misinformation, in order to create knowledge in a world where information manipulation is becoming a weapon, started during the MindBugs project. The results of this project were presented internationally at different venues like IRCAM-Paris, KI Park-Berlin, and Starts4Watter - Brusseles. We also had a dedicated venue in Cluj-Napoca where visitors could interact directly with the tools and games developed. All of these would not have been possible without the support of MediaFutures under the framework Horizon 2020.

Building forward on the MindBugs universe came MindBugs Discovery. The project was designed together with media specialists during Media Blend Hackathon in Vienna and presented at International Press Institute (IPI) World Congress. Because of the interest it sparked, the project became more than a discussion. We, as a team, are grateful for the support offered by IPI.

I would like to express my appreciation to the European Union's Horizon 2020 research and innovation action programme, and in particular to the AI4Media organization. This support has provided me the opportunity to collaborate with experts in the media field, understand their challenges and try to develop a tool aimed at enhancing democratic processes and community well-being.

The MindBugs Discovery has indirectly received funding from the European Union's Horizon 2020 research and innovation action programme, via the AI4Media Open Call #2 issued and executed under the AI4Media project (Grant Agreement no. 951911).



Contents

Chapter 1

Introduction

In the last year, we, as humanity, have created the technology to pass the Turing test. If one is optimistic, is possible to say that humans and machines have the same capacity of writing quality content, but if you take a more realistic approach, machines are better. Despite the big advancements with both their threats and opportunities, we still do not understand how deep learning works. It can be said that we created the alien technology that we feared for so long: it surpasses us in terms of intelligence (defined as the ability to solve tasks) and understands and exploits us, but humans can not understand it.

In the context of this new technology, a flood of text AI bots could mean the end of democracy, a flood of generated art could mean the end of knowing human creativity and a flood of generated music could mean the end of understanding someone's feeling but an exploitation of what we like [Tho23]. We already witnessed increased problems due to the recommendation algorithms of social media, which influenced the ability to learn and focus on millions of people [Sub18]. I strongly believe that at this point we can not imagine the implications and effects of the new technology on humans.

As Frank Herbert said [Her87], the reformers have always caused more damage than any other force in human history, when they try to completely change the course of things. On the contrary, one should focus on finding the natural course of things, follow its trajectory, and influence it in a positive way. In this line, the powerful new AI technology is here to stay, and despite the threats, we need to use it, understand it, and create a better world.

In contradiction with the concerned beginning, I strongly believe that AI could be the best and most interesting technology developed by humans so far. It is the most advanced intelligence we created to solve different kinds of problems. It is used not only to empower individuals but also to reduce suffering in key areas where human intelligence seems to reach its limitation or move too slowly. It can already improve day-to-day life in a meaningful way, accelerate the development of

drugs, create simulations for cancer patients with different types of treatment and the list gets longer and longer.

In this line of thought the importance of Explainable AI (XAI) is growing. How can we create systems that use the power of deep learning, while also allowing humans to understand most behind the scenes and influence their behavior? These kinds of systems are important in order to expand the current use of deep learning to critical areas such as journalism or medicine while keeping the responsibility and decision on the human side. One of these initiatives is the tool presented in the following pages: MindBugs Discovery, a tool for fact-checking journalists.

The MindBugs Discovery aims to create an interactive environment for visualizing the hidden structure of misinformation. In a world where reality and fiction merge, a scalable system with verified information is beginning to be a necessity. The tool is designed for journalists to help them ease their work and also for the general public who has the curiosity of looking into disinformation techniques and propaganda. It is an initiative that wants to gather the sparse work of fact-checking organizations into a single place, in order to help people understand and easily identify the fake information which they encounter. It is a centralized space and a search engine, where the work and knowledge of human specialists are provided.

MindBugs Discovery is in line with my previous efforts in regard to XAI. Both in the present work but also in "The Profile" [CG23], Transparency [Che22] and MindBugs AR Game [Min] the question from behind the scene remains the same: how can deep learning empower all people, not just professional or companies developing it? How can we influence society in retaking control and understanding how AI impacts day-to-day life?

"The Profile" is an art installation that uses deep learning to create an immersive experience where users can explore the capabilities of AI. This work is aimed at impacting emotionally the users, trying to make them understand, on a both rational and emotional level, the ease of generating fake video content and the impact on the people who may become victims of these technologies. This line of work was further developed with "Transparency", another AI installation exhibited at "MindBugs AI & Art Exhibition " in Cluj-Napoca in September 2022. Inside the exhibition, visitors were taking a selfie and in less than 2 minutes, they were projected in the middle of the exhibition saying fake statements. Both works rely on the emotional impact to send a message regarding the changing nature of the video evidence that we encounter online.

The MindBugs[Min] project comes to complement the image generation direction presented in "The Profile" and "Transparency". Along with the start of the war in Ukraine and the apparition of ChatGPT, the internet was flooded with an abundance of misinformation and disinformation. The first project in this direction

was MindBugs AR Game, which uses AI to analyze fake information and extract patterns. These patterns are further visually interpreted by artists and incorporated inside the mobile game in the form of MindBugs. The game is designed for the young generation and is a gamified manner in which young people are presented with big data about fake news. It represents an effort to help the young generation fortify their minds by being trained to spot cognitive biases and characteristics of fake news.

Built further on the projects presented above, MindBugs Discovery is a tool for experts. It unites the efforts of multiple fact-checking organizations under the same system, trying to become a reliable search engine in a safe information space.

The tool contains a web-based user interface (UI) which allows the user to explore the structure of fake information. It is designed in the form of a 3-dimensional connected knowledge graph and has the functionality of clicking, dragging, and analyzing different nodes, which represent fake statements or key information associated with them.

The MindBugs Discovery tool provides users with the capability to enter data using natural language. After doing an analysis of the data using its intrinsic keywords and entities, the algorithms proceed to conduct a comprehensive search for existing fact-checking activities that are in line with the provided material. This system makes significant steps towards reducing redundant journalistic efforts, therefore acknowledging and appreciating the previous contributions made by other fact-checking specialists.

The MindBugs Discovery project has made several distinct unique contributions. Primarily, the system consolidates data from several fact-checking groups, so integrating disparate efforts into a single repository. The project integrates a scalable architecture that is particularly engineered to facilitate the extraction, loading, and transformation of data, increasing the resilience and flexibility of the system. Moreover, the system utilizes an innovative combination of search algorithms in order to maximize the retrieval and analysis of data, improving misinformation monitoring. The initiative introduces a novel approach to data collection that offers significant advantages for refining large language models such as GPT via community analysis. The search engine of the tool is improved using a combination of Fast Random Projections and k-nearest Neighbours.

Finally, the system presents a novel technique for data display to the user. Instead of just presenting a catalog of results, MindBugs Discovery utilizes visualization methods to emphasize the connections between data points, providing a justification for the outcomes it produces. This facilitates a broader and more intuitive comprehension of the data, hence augmenting user involvement and fostering the development of analytical thinking skills.

Chapter 2

Knowledge graph fundamentals

Knowledge graphs have been experienced for a long period of time in multiple setups and with different approaches [JPC+21]. Using a graph-based abstraction to capture knowledge has multiple benefits when compared with the usual approaches like NoSQL. They provide a simple and intuitive representation of multiple domains, where edges and paths represent complex connections between various entities[HBC+21]. In addition, the structure of the knowledge is postponed, allowing the systems to grow in a more flexible manner. Knowledge graphs began to gain popularity in 2012 when this concept was integrated into Google Search Engine[JPC+21].

According to a Google, blog [Sin12], the primary motivation behind Google's decision to integrate knowledge graphs into their search engine is to accurately capture the user's intention, understand concepts as entities rather than mere strings, and deliver additional results that are of interest for the user. In contrast with the previous version, which was solely based on string matching, the new functionalities powered by knowledge graphs could understand the concept of multiple distinct entities identified by the same string representation. Also, by leveraging the relations captured within the knowledge graphs, the search engine can generate a concise summary of the queried element, resulting from understanding the information from the web. When users search for a specific relation, the search engine can also identify other connected nodes with similar relations, thereby offering additional entities and relationships that are more likely to be of interest to the user.

The most used definition of a knowledge graph is that they are a multi-relational graph composed of entities and relations. This graph is intended to contain and convey knowledge of the real world. Its nodes represent entities of interest and edges contain potentially different relations among these entities [HBC+21]. The knowledge graphs can be open like BabelNet, DBpedia, Wikidata, and many others, having the purpose of creating a substrate of common knowledge, making their content accessible for the public good. On the other hand, enterprise knowledge

graphs are used internally by a company to create profit.

Because of their flexibility sometimes is hard to evaluate the quality of a knowledge graph. One way to achieve this is to use shape graphs, which define a set of constraints for each node. Boolean combinations of shapes can be defined using boolean algebra, and after completing this step a conformance metric according to it can be computed. A node conforms to a shape if it satisfies all the constraints imposed, and using this measure it can be assessed that a graph is valid or invalid in accordance with a certain shape or set of shapes. There are two main shape languages proposed for RDF graphs: Shape Expressions (ShEx) and Shapes Constraint Language (SHACL)[HBC+21].

2.1 Data models for knowledge graphs

Because graphs provide a flexible way to store and integrate diverse and sometimes incomplete data, there are multiple data models used in practice[HBC+21]. Each of the data models has its particularities, which allows it to be more efficient for a particular task. The main models are presented in Figure 2.1 and detailed below:

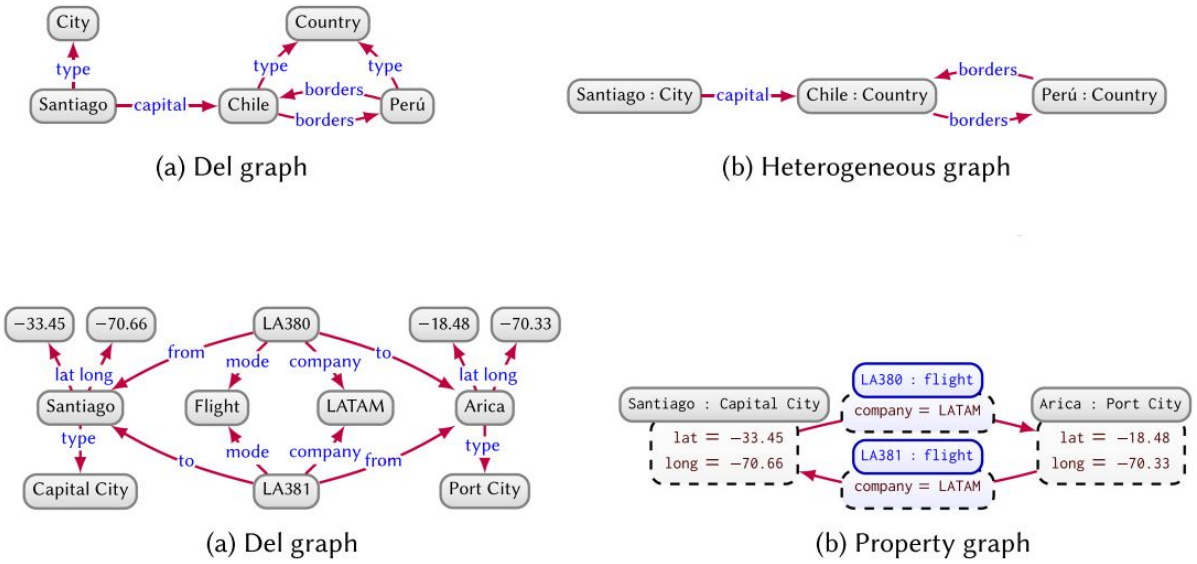


Figure 2.1: Representing data in knowledge graphs [HBC+21]

1. Direct Edge-labelled graphs: in this type of data model, nodes represent entities, while edges represent binary relations. Adding data to such a graph usually involves adding new nodes and edges. A standard data model used for this type of graph is the Resource Description Framework (RDF). RDF defines three types of nodes: Internationalised Resource Identifiers (IRIs), literals, other data types, and blank nodes used for the existence of an entity.

2. Heterogenous Graphs: each node and edge is given one type. They are similar to direct edge-labeled graphs, but in this setting, the type of node is part of the graph itself. It allows for both homogenous and heterogenous relations. They typically assume a one-to-one relationship between the type and the node.

3. Property graphs: it allows for a set of property-value pairs and a label to be associated with different nodes and edges, adding more flexibility in representing the data in the knowledge graph. This type of representation is not yet standardized, but they are used in multiple popular graph databases.

4. Graph dataset: allows for managing several graphs and contains a set of graphs and default graphs. Since knowledge graphs can be used to query multiple information sources, each one can be represented as a knowledge graph and then they can be organized in the form of a graph dataset. The default graph is referenced by default, as the general base of knowledge.

A first attempt to represent a knowledge graph was to use one-hot-encoding, generating a vector of length N x V[HBC+21]. This would result in very sparse data structures, which is detrimental to most algorithms and machine learning models. Due to the advancements in representing knowledge graphs, in general, their types and applications can be split into four main categories. The main categories are illustrated in the Figure 2.2 [JPC+21]:

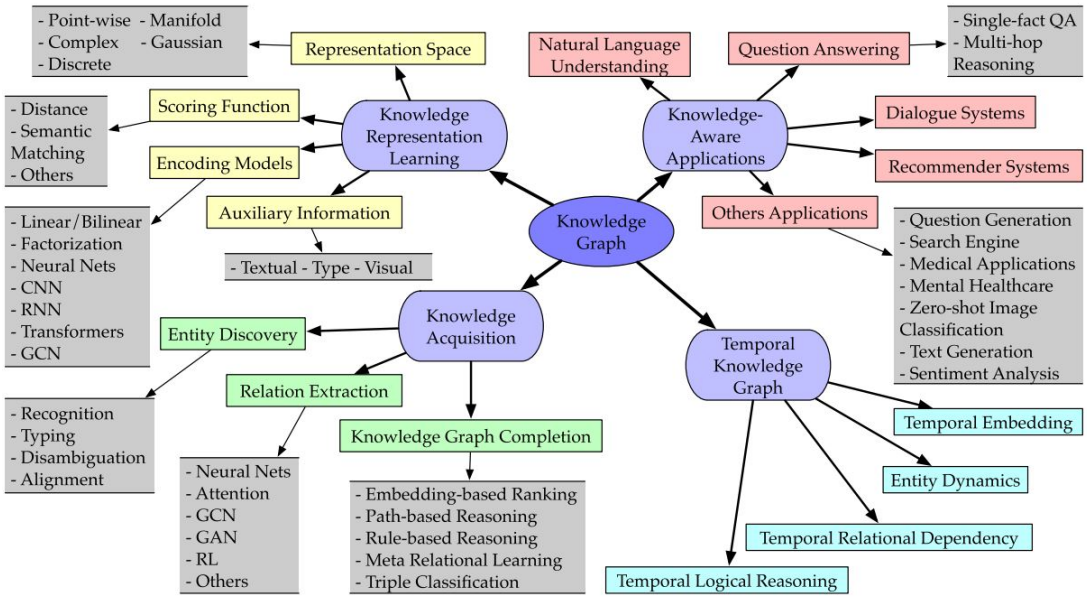


Figure 2.2: Knowledge graphs categories [JPC+21]

2.2 Knowledge representation learning

Knowledge representation learning is a statistical learning approach. It contains a representation space, where the nodes and edges are represented, a scoring function for evaluating the resulted connections, encoding models for learning interactions, and auxiliary information that can be incorporated. The main goal of embeddings is to represent knowledge in a low-dimensional space which can be used more easily and efficiently by other algorithms. Most commonly, the embeddings define a scoring function that starts from the node and edge and computes the plausibility of the edge [HBC+21].

2.3 Knowledge acquisition

Knowledge acquisition is a method that aims to construct knowledge graphs from unstructured text or other structured or semistructured sources. The main components of knowledge acquisition are relation extraction, knowledge graph completion, and other entity-oriented acquisition tasks. Knowledge graph completion is developed in order to add new triples to a knowledge graph. Entity and relation predictions are made using embedding-based models which rely on ranking algorithms, path reasoning using relations and rules, and deep learning methods[JPC+21].

Entity discovery tasks are split into entity recognition, which focuses on tagging the named entities from various texts, and entity disambiguation which connects entities named partially or with different names to the same base entity. The last step is usually called Entity Linking, and most commonly it searches Wikipedia and it checks if two words have the same Wikipedia page. For example, the page "Napoleon" from Wikipedia is the same as "Napoleon Bonaparte".

Relation extraction is an important task in identifying unknown connections in a text using weak supervision. While traditional approaches rely heavily on feature engineering, deep learning models are revolutionizing the representation learning of knowledge graphs and texts. Neural relation extraction employs different neural networks like CNN, multi-instance learning, and PCNN. In order to include word-level attention, multiple variants of attention mechanisms are employed to capture different nuances of semantic information of words or to reduce the noisy inputs.

2.3.1 Ontologies and interpretations

In addition to the knowledge graph representation, ontologies and rules can be used to define and reason using the information encoded. Knowledge graphs can contain explicit statements like: "Bucharest is the capital of Romania", but quantified state-

ments like "All capitals are cities" are better represented using ontologies and rules. The combination of the two approaches can be used to gain extensive knowledge of the available data. Deductive methods can be employed to obtain information like "Bucharest is a city", while inductive methods may be used to extract additional knowledge from the graph itself [HBC+21].

An ontology is a concrete, structured, and formal representation of knowledge, what the terms mean within the scope they are used, how they are organized and what is their hierarchy. One of the most popular ontologies is the Web Ontology Language (OWL), which is compatible with RDF graphs[HBC+21].

Humans interpret the knowledge and connect the information from the knowledge graph to real and existing entities. The real-world entities are known as the domain graph, and the process of interpretation involves mapping the knowledge graph nodes and edges to real-world entities and relations. After the interpretation phase, different queries can be answered by considering different assumptions:

- Closed World Assumption(CWA): this assumption considers that everything that is unknown, thus not presented or found in the knowledge graph, is by default false.
- Open World Assumption(OWA): this assumption states that it is possible for a piece of information to be true even if it is not presented in the knowledge graph.
- Unique Name Assumption(UNA): states that no two nodes can map the same entity.
- No Unique Name Assumption(NUNA): there can be multiple nodes that refer to the same entity.

Beyond these base assumptions, a knowledge graph can enforce multiple semantic conditions like sub-property, property, category of etc. These conditions then form features for an ontology language. It is possible to then associate classes to propertied by defining their domain and range. Some of the most used types of properties are transitive, reflexive, symmetric, asymmetric, and irreflexive.

Wikidata

Launched on October 30, 2012, by Wikimedia Deutschland, Wikidata is an open, cooperative initiative that has experienced consistent growth in popularity. It is backed and maintained by the Wikimedia Foundation [VK14]. Wikidata's core objectives are dual-pronged: firstly, it functions as the central repository for the structured data of all its sister projects under Wikimedia, including Wikipedia. This helps

in avoiding information duplication and contradictions while supporting multilingual capabilities and management. Secondly, it supplies data to external projects and initiatives, while also accommodating intricate queries on the present knowledge base.

Wikidata's function extends beyond merely storing facts. It also holds the associated reference sources, enabling data validation and timeline creation. For instance, changes in a country's population can be referenced to the census and tracked over time.

Labels, aliases, and descriptions of entities in Wikidata are offered in over 350 languages. The fundamental framework of Wikidata comprises items (with a label, a description, and numerous aliases, collectively referred to as terms), properties, and values. These are interconnected in statements that are similar to an RDF triple, although Wikidata's statements can be slightly more complex as they may contain qualifiers (for further context) and references (to substantiate the claim).

As of October 2018, more than 60,000 registered authors (contributors who have made ten or more edits), along with anonymous users and automated bots, have contributed to over 53.5 million data items in Wikidata.

2.4 Temporal knowledge graphs

Temporal knowledge graphs are an approach to incorporating the temporal dimension into the knowledge graph representation. It includes temporal embeddings, temporal relational, temporal reasoning, and many others.

2.5 Knowledge graphs applications

Knowledge graph applications include natural language understanding for different smart assistants, question-answering, recommendation systems, and many others. The world of applications based on knowledge graphs is continuously expanding since the advancement of hardware and algorithms, allowing developers to create useful devices and tools for both specialists and general users. In the last period of time, multiple language representation learning systems were developed. They use a combination of both knowledge graphs and deep learning methods in order to create a comprehensive and easy-to-use representation of a language model. Knowledge graphs are also heavily utilized in question-answering systems. They are used to answer simple facts and are enhanced with different pruning systems in order to reduce the search space. Another area in which graphs are widely used is multihop reasoning, where utilizing the structured organization of knowledge graphs system

can improve the commonsense fusion of different systems. In order to solve the sparsity and multitude of online information, the recommender systems are based on knowledge graphs to enable users to navigate more easily in their area of interest.

2.6 Searching in knowledge graphs

Searching in knowledge graphs has emerged as an essential tool for navigating interconnected data and extracting relevant information. In contrast to traditional databases, which rely on tabular structures, knowledge graphs represent information as nodes (entities) and edges (relationships), simulating the complex web of real-world associations. This graph-based structure facilitates more intuitive queries, allowing users to investigate connections, infer new relationships, and retrieve contextual data. This exploration is facilitated by algorithms spanning from depth-first and breadth-first search to more advanced techniques such as SPARQL queries and graph neural networks. As the volume and intricacy of data increases in the current digital era, the ability to search within knowledge graphs efficiently and precisely becomes crucial, providing nuanced insights that are frequently unattainable with traditional data models.

There are multiple languages that deal only with querying knowledge graphs like SPARQL, Cypher, Gremlin, and G-CORE. The common primitives that stay at the base of these languages, which are also the primitives of querying a knowledge graph are[HBC+21]:

1. Graph Patterns: the terms in the graph are divided into constants and variables which are prefixed with a question mark. The pattern is then evaluated by mapping the variables to the constants from the graph structure.

2. Complex graph patterns: allows the tabular results from multiple graph patterns to be processed using relational algebra including operations like FILTER, UNION, LEFT JOIN etc. The complex graph pattern can combine the results of simple patterns in order to show more structured results.

3. Navigational graph patterns: it tries to find paths, which are composed of edges and nodes between, two different nodes, a node and a path or two paths. This allows for finding correlated entities or solving different problems using the knowledge graph representation. Since if a cycle is present there can be an infinite number of results, so most navigational graph patterns have different constraints like no repeating nodes, only the shortest paths or no repeating edges.

SPARQL

At its core SPARQL is a query language designed for matching graphs. For a data source labelled as D, a query is formed from a pattern that is matched against D, with the answers derived from the resulting values. This data source D can be pulled from numerous sources. A SPARQL query has three key components[PAG06].

The first part, pattern matching, offers several unique attributes including optional elements, pattern unions, nested patterns, value filtering for potential matches, and the choice of data source for pattern matching.

The second component, known as solution modifiers, allows modification of values from the outcome of the pattern matching (represented in a table of variable values), by implementing traditional operators such as projection, distinct, order, limit, and offset.

The final part is the output of the SPARQL query, which can take on a variety of forms: confirmation or denial queries, selection of variable values that match the patterns, generation of new triples from these values, or descriptions about queried resources.

SPARQL is the query language used to retrieve data from Wikidata. Wikidata is a free and open knowledge base that contains structured data from Wikimedia projects and other sources. Creating a SPARQL Query[Wik23]:

- Select Clause: This clause determines what the output of your query will look like. Example: SELECT ?item ?itemLabel
- Where Clause: This clause specifies the pattern that you want to match in the database. Items in Wikidata are prefixed with ”wd” and properties with ”wdt”. Each type or subtype of item an property has a specific code and can be found on Wikidata.

  Example: WHERE ?item wdt:P31 wd:Q146 . . This matches items that have the property ”instance of” (P31) ”cat” (Q146).
- Service Clause: This optional clause is used to retrieve additional data such as labels for items. Example: SERVICE wikibase:label { bd:serviceParam wikibase:language ”[AUTO_LANGUAGE],en”. }

2.7 Rules and reasoning

Rules provide a simple and intuitive way of implementing reasoning. Rules are composed of the head(IF) and body(THEN). A rule means that if one can replace the variables of the body with terms from the data graph and form a subgraph, then using the same replacement of variables in the head will lead to a true conclusion.

Inductive reasoning is used to generalise knowledge from existing information in the knowledge graph. This new knowledge can potentially be imprecise or even erronated. It is important to also compute confidence for predictions when performing reasoning on the graph, and these predictions represent encoded patterns existing in the graph. The knowledge can be obtained from the graph using supervised or unsupervised methods. Supervised methods learn a model from a set of example inputs and their labelled outputs. To avoid the expensive process of manually labelling the data, some methods create automatically labelled data which is then fed into a supervised model. This approach is known as self-supervision. In contrast, unsupervised methods do not use labelled data, but apply a function which is usually of a statistical nature, to map inputs to outputs. In the case of unsupervised learning, there is an entire domain that studies graphs analytic in order to detect communities of clusters, find central nodes and edges and so on.

2.8 Graph Analytics

Graph analytics is the application of analytical algorithms that study the topologies of graphs, how nodes are connected, groups categories and patterns.

- Centrality analysis is a direction that studies in depth the identification of the most important (”central”) nodes and edges. Specific measures used to determine node centrality include degree, betweenness, closeness, eigenvectors and others. Node centrality allows for the determination of the most connected entities, and edge centrality to the most common and important properties and relations.

- Community detection allows for the detection of entities with a high level of connectivity. It looks for communities that are resilient through the multitude of edges, and it includes, among others, measuring graph density, k-connectivity and computing spanning trees.

- Node similarity aims to find nodes that have a high degree of similarity. Node similarity metrics can be computed using structural equivalence, random walks, diffusion kernels and so on.

- Graph summarization: aims to provide a birds-eye view of a knowledge graph by extracting the high-level structure of a complex graph. The methods are based on quotient graphs, where nodes in the input are merged while preserving the edges.

2.9 Graph database - Neo4J

Neo4j is a highly scalable, native graph database platform that is designed for leveraging data relationships in order to derive insights. It's based on the property graph model and utilizes Cypher, a declarative graph query language that allows for efficient and expressive querying and updating of the graph store. Neo4j is acclaimed for its speed, flexibility, ease of use, and capacity to handle complex queries with extensive join operations, even with large amounts of data. It is used across a variety of applications, including recommendation systems, fraud detection, real-time graph analytics, network and IT operations, and identity and access management, to name a few. As an open-source project, Neo4j has a vibrant community of users and developers, and it can be employed in both on-premise and cloud-based environments. Neo4J has both a cloud version and a desktop one. It also provides access to multiple libraries for Data Science.

Previously Neo4J was used in multiple recent types of research:

- Life cycle inventory [HN22]: improving detailed analytics for quantifying and predicting the impacts on the environment of different products and services.
- Lung cancer graph [Tuc22]: A new graph data model has been developed for non-small cell lung cancer clinical and genomic data. The model serves two main purposes: (1) to foster efficient graph analytics within the Neo4j framework or via tools compatible with existing Neo4j APIs, and (2) to lay the groundwork for a model that can be expanded to accommodate other types of cancer and additional datasets, including those obtained from electronic health records and other real-world data sources. The graph is mainly used at 2 different levels: creating networks between patients based on similarity scores and genomic features and biological networks within single patient samples.
- Analysis of film data [LHS17]: Analyzing film data is critical for websites to identify connections within the data. Instead of traditional relational databases, a modern and increasingly popular graph database, Neo4j, employs graph-based principles to characterize data models, easily transforming data into nodes, edges, and their corresponding attributes.

After reviewing different studies and observing the remarkable capabilities of Neo4j in a variety of applications, it is clear that this technology is useful for different tasks and for showing underlying structures of data. Graph databases represent a revolutionary change in data management and analytics. Neo4j is an emerging technology that is well worth investigating due to its expanding use across industries and the growing number of solutions that leverage its graph database capabilities. It quickly

becomes a very useful tool in the era of big data due to its capacity to efficiently manage complex relationships.

Chapter 3

MBD System architecture

The MindBugs Discovery platform is created to interface with multiple fact-checking websites, gather data in the form of a knowledge graph, interact with external APIs, and provide users with the possibility to explore the interconnected landscape of misinformation. Given the multifaceted nature of its operations, the system necessitates a distributed architectural framework. This design ensures that dedicated software modules are allocated to more simple tasks, enhancing the reliability, scalability, and overall performance of the platform.

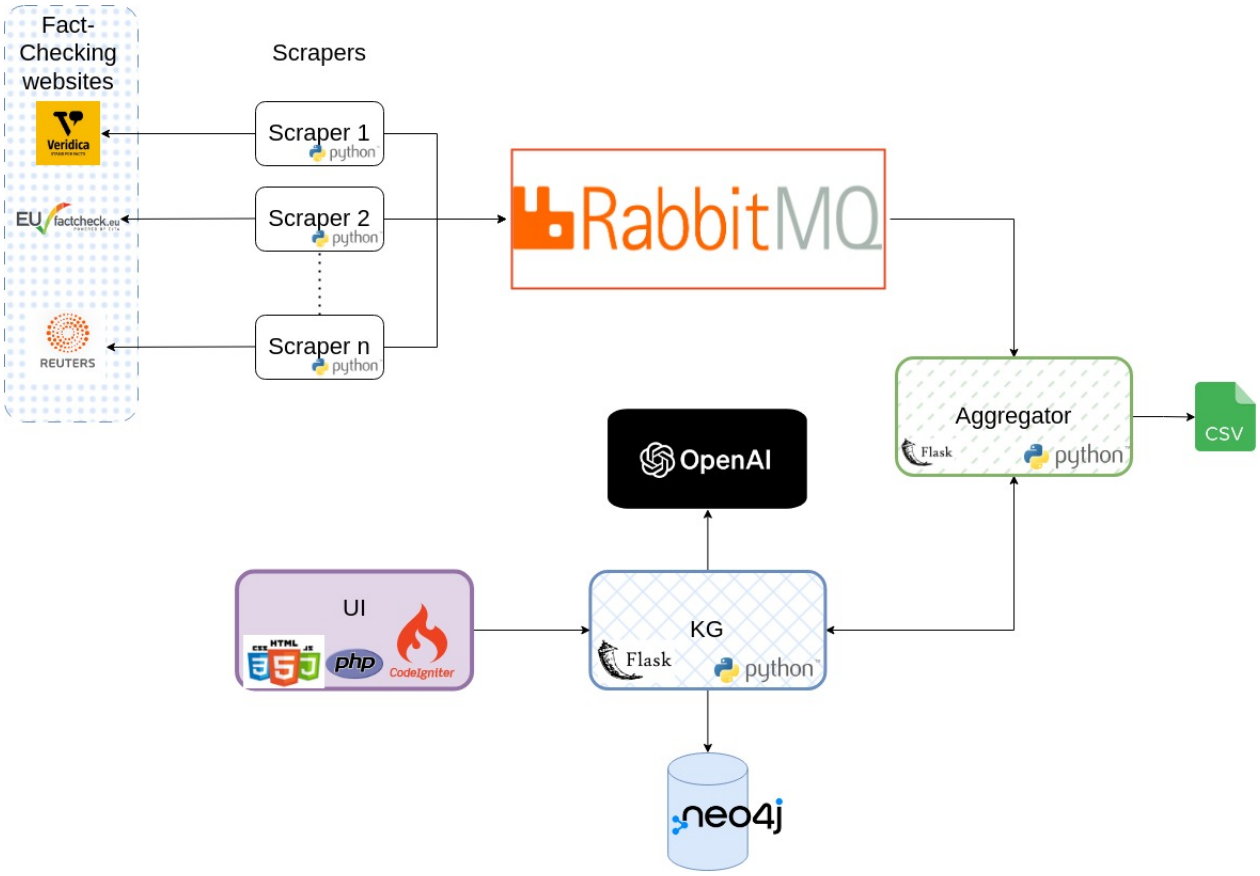


Figure 3.1: System architecture

Using different scarpers the system collects data from different fact-checking organizations. The data is passed through the RabbitMQ queue to the Aggregator. The Aggregator analyzes and stores all the data, extracting structured information like the entities involved, keywords, or locations. The data is further structured and stored by the KG component in the Neo4j graph database. There, the data is queried and analyzed using different algorithms for community and similarity detection. Also, the KG component is the point where the system communicates with OpenAI API. The KG component is called by the UI component, where the system interfaces with the users.

Scrapers

In this system, specialized web scrapers are developed to collect analytical and text data from various fact-checking websites. These scrapers are tailor-made for each source, ensuring that the data collected is correct and valued to its maximum capacity. A generic scraper would have been capable of collecting larger volumes of data, but it would have been more prone to errors and missing important information. In this context, the accent is put on the reliability and quality of the data, more than on the volume.

Once the data is acquired, it is channeled into a RabbitMQ queue. The queue has the purpose of decoupling the scraping modules from the central aggregation unit. This layer of abstraction improves the system's reliability, making it easier to scale up the number of scrapers or modify them independently.

Aggregator component

The central aggregation unit, called the "Aggregator", is responsible for assimilating the data gathered by the scrapers. This module normalizes and further analyzes the text, extracting important information such as keywords and entities involved. Here is the place where the data mining on the raw inputs is done, in order to create a structured dataset of fake statements.

All the data received, both in raw format and after the processing, is stored inside CSV files. This precautionary measure is done in order to be able to change the algorithm or data processing techniques, without having to re-scrape the websites, increasing the decoupling from the scrapers. Once the Aggregator receives new data, it calls the "KG" component to add the new entries to its database. The Aggregator component has an API created in Flask to allow the KG component to analyze with its algorithms the information inputted by the user.

KG - Knowledge Graph component

The component responsible for the knowledge graph creation is called ”KG”. This is the main component that interfaces between the user and all the other reachable micro-services. It communicates with the Aggregator component for analyzing data of the new information introduced by the user. It also has the responsibility of communicating with the OpenAI API in order to generate text according to the user's needs.

Once new data from the Aggregator reaches the KG component, it is decomposed and stored as a knowledge graph. The component uses Neo4J, a graph database management system designed for storing and querying highly interconnected data.

When the users query the system, the KG component performs searches in conjunction with the FastRP and K-Nearest Neighbors (KNN) algorithms from Neo4J, ensuring that the data is reliable and the process is fast. After gathering the most similar nodes and the detected communities, a suggestion for the debunking article is generated using the gpt-3.5-turbo model. The information obtained by the system is then returned to the UI component to be shown to the user.

User Interface

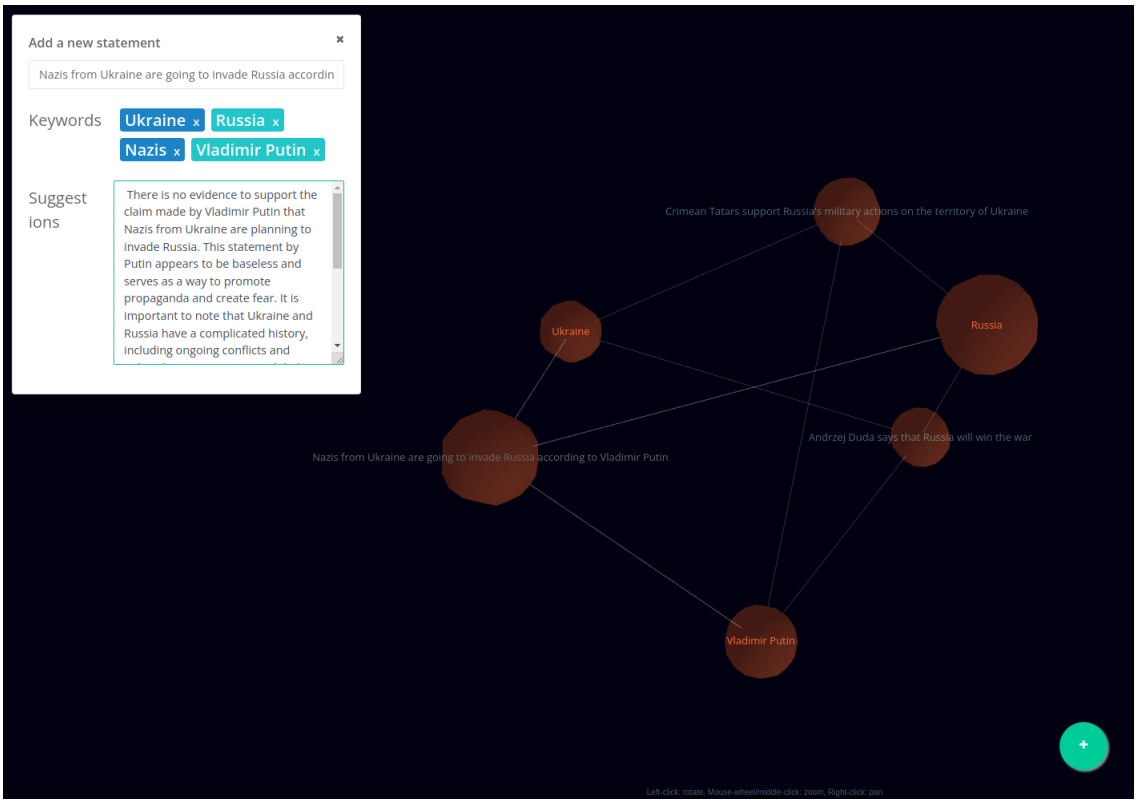


Figure 3.2: UI overview

To facilitate human interaction with this data, the system features a user-friendly

web interface built on the CodeIgniter PHP framework. The interface employs the force-three-js [Thr23] library to create interactive, three-dimensional knowledge graph visualizations, allowing for an intuitive exploration of the interconnected realms of misinformation.

The knowledge graph provides end-users with an interactive experience. Users have the capability to manipulate the graph by rotating it to obtain other viewpoints, adjusting the magnification level to examine details or obtain a broader overview, and manipulating nodes by tactile means, allowing users to physically manipulate and reposition individual nodes within the graph.

Using the UI, users can input new fake statements that they encountered online. Upon submission, the system integrates this new data into the existing knowledge graph, as shown in Figure 3.2. The platform contextualizes the newly introduced statement and also identifies historical precedents or analogous cases within the database. Furthermore, it provides data-driven recommendations to guide the user in writing a comprehensive and insightful debunking article. This robust functionality transforms each user into a proactive participant in the fight against misinformation while enhancing the utility and depth of the knowledge graph.

Chapter 4

Knowledge acquisition

Ensuring data reliability is of importance in the MindBugs Discovery system, particularly given its target audience of professional journalists who require accurate and credible information. To uphold this standard, our data acquisition process is designed to crawl only verified fact-checking websites. For each fact-checking website, a custom scraper is designed in order to maximize the quality and quantity of the information extracted.

The architecture of the MindBugs Discovery acquisition data is split into two distinct types of micro-services: specialized web scrapers and a data queuing mechanism that serves to channel information to a centralized aggregation point. The system and its organization are shown in Figure 4.1. This modular approach enhances the system's stability, allowing each component to operate autonomously while still contributing to the overall functionality.

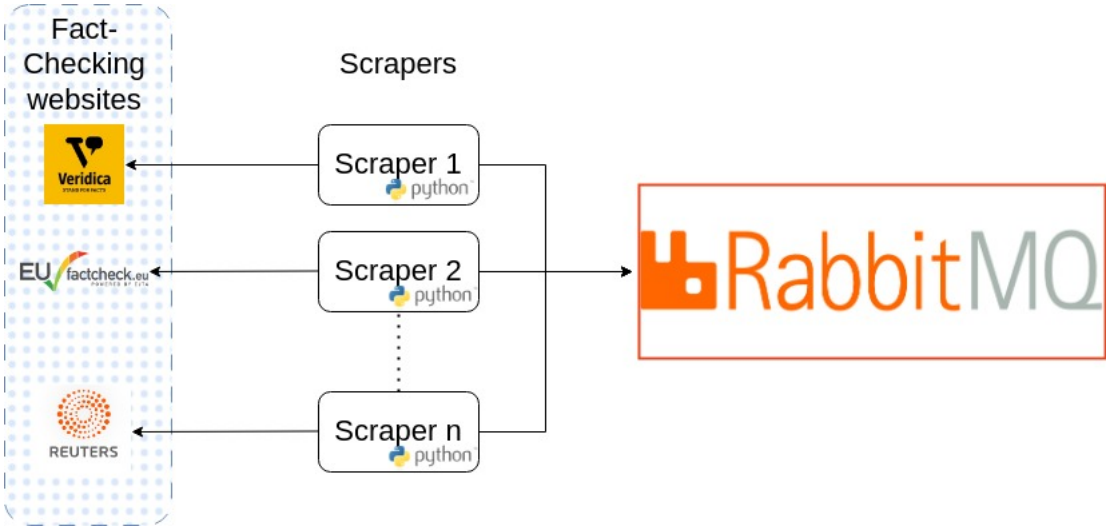


Figure 4.1: MindBugs Discovery knowledge acquisition

4.1 Web scrapers

The dedicated web scrapers serve the role of gathering data exclusively from verified fact-checking platforms. Each scraper is tailored to interface with a specific website, ensuring the maximum extraction of structured information. To facilitate optimal efficiency and maintain data integrity, each scraper has an internal state management system that logs already-parsed web pages. To minimize the computational burden on the targeted websites and adhere to best practices, these scraping operations are scheduled to run once daily.

The web scraping components employ the BeautifulSoup library[Ric07], a tool for parsing HTML and XML documents. Utilizing BeautifulSoup enhances the precision and efficiency of the scraping process, allowing for the effective extraction and manipulation of complex data structures.

Since each website has a unique structure, the web scrapers are designed to look for the specific information structure of the site in question. This means identifying and parsing HTML elements such as 'div' tags associated with specific CSS classes, text formats, and other distinguishing markers. In the case of Veridica, the scraper not only seeks out designated 'div' classes but also searches for uniquely formatted text such as "stire:" or "naratiune:" to ensure precise and comprehensive data extraction.

The web scrapers are designed to extract all available fields from newly published debunking articles. Following the data extraction phase, these scrapers interface with a specialized class, known as the "QueueConnectionModule", to facilitate the transmission of the data to a designated queue. This architecture ensures both robustness and data integrity in the system's data aggregation pipeline.

In the initial phase of this project, our focus was on the 'Veridica' website, which was selected as the initial data source for our web scraping efforts. Veridica is a publication specializing in monitoring, analyzing, and dismantling fake news, disinformation, and manipulation campaigns in Central and Eastern Europe. Veridica project was launched in the context of the health crisis and the infodemic, it is undertaken by the International Alliance of Romanian Journalists and carried out together with journalists, experts, and researchers from Central and Eastern Europe.

4.2 Queue

The queueing mechanism employs RabbitMQ, a leading message-brokering service known for its robustness and scalability. RabbitMQ serves as the intermediary layer, providing a decoupling interface between the web scrapers and the central data aggregator.

This design choice is advantageous for system reliability. The architecture allows for an arbitrary number of scrapers to be in operation concurrently, each tailored to a specific source. Upon gathering the requisite data, each scraper interacts with the RabbitMQ queue through a specialized API. This queued data is then consumed by the central "Aggregator" for further processing and analytics. By using RabbitMQ, the system gains significant advantages in fault tolerance, load distribution, and overall reliability.

Chapter 5

Knowledge graph construction

The software application handles each false statement from the dataset and integrates it into a comprehensive knowledge graph. Each statement is added as an individual node within this graph, effectively becoming a distinct point in the network.

After successfully inserting the false statement, the program then proceeds to identify and incorporate associated keywords and entities. These includes important terms within the statement or recognizable elements like names, locations, or organizations, for instance.

Each keyword and entity is added as its own node in the knowledge graph and is linked to the false statement from which it was derived. These links or 'edges' in the graph represent the relationships between the statements and their associated keywords and entities.

These connections are used to explore the underlying structure of our data, highlighting the commonalities and patterns across different false statements. By visualizing these points of connection within the knowledge graph, we gain a clearer understanding of the nature and spread of these falsehoods, enhancing the ability to analyze and address them.

The method for integrating the fake statements into the knowledge graph is presented below and shows how the system introduces at first the statement, then its associated entities and lastly the keywords.

```
for index, row in df.iterrows():
    connector.insert_statement(row)
    connector.insert_statement_entities(row, entities)
    connector.insert_statement_keywords(row, entities)
```

5.1 Data storage in Neo4J

Once the data has been processed and essential information has been extracted, the fraudulent statements, their associated keywords, and entities are stored in the Neo4J database. This strategy assures a well-structured, easily accessible data repository that can be effectively analyzed for patterns and insights.

Neo4J is chosen as the preferred database for this operation predominantly due to its distinctive features and capabilities, as described below:

- As a graph database, Neo4J excels at storing, managing, and querying data that is highly interconnected. The relationships between various fraudulent statements, their keywords, and associated entities are as essential, if not more so, than the data points themselves, so this characteristic is especially pertinent in this context.

- Built-in support for machine learning algorithms, such as the Louvain algorithm for community detection, K-Nearest Neighbors (KNN) for similarity measurements, and FastRP for node embeddings. These algorithms are not only well-implemented but also optimized for performance, enabling more efficient data analysis and feature extraction. The availability of these advanced capabilities lets the programmer build on top of the algorithms, focusing on the quality of data and the methods used.

- Performance and Efficiency: Neo4J's capacity to execute complex queries quickly and efficiently enables us to manage large quantities of data without sacrificing performance. This efficiency is a result of Neo4J's fundamental design, which enables data to be directly linked, reducing the need for expensive join operations typical of traditional relational databases.

- Neo4J's proprietary query language, Cypher, is designed specifically for querying graph data. The declarative nature of Cypher enables us to concentrate on what to retrieve rather than how to retrieve it. This enables a more intuitive and sophisticated analysis of the relationships between data points.

- Neo4J is designed for scalability in terms of both data volume and complexity. This indicates that Neo4J will be able to manage the increased load and continue to provide valuable insights as the dataset of fraudulent statements grows and evolves.

- Community and Support: Finally, Neo4J has a robust community and voluminous documentation, which provide valuable support and resources when working with the platform. This solid support can be crucial for addressing

any challenges or complexities that may arise when dealing with a dataset of this complexity.

In conclusion, the decision to utilize Neo4J is based on its capacity to manage complex, interconnected data efficiently, execute queries quickly, scale to accommodate expanding data volumes and provide strong community and documentation support. These factors make Neo4J not just a suitable option for this operation, but a highly effective one.

5.2 Inserting statements nodes

Since each fake statement represents one node the query for inserting the statements into the knowledge graph is straight-forward. The code uses noe4j python library to connect with the Neo4J database. It uses Cypher query language to enter the statements as nodes and to set some specific attributes of the node: id, statement, and full explanation. After the insertion of the statements nodes, a knowledge graph with sole nodes as statements are obtained, as depicted in Figure 5.1.

```
def insert_statement(self, row):
        INSERT_STATEMENT = """USE news
            UNWIND $inputs AS row
            MERGE (n:Fake_Statement {id: row.id})
            SET n.statement = row.statement,
            n.text = row.full_explanation
        """
        with self.driver.session() as session:
            session.run(INSERT_STATEMENT, inputs=row.to_dict())
```

5.3 Inserting entity nodes

Every false statement from our dataset is analyzed to extract the entities involved in it. These entities serve as subjects, or main points of focus, in the fake news narrative. Once identified, these entities are parsed and inserted into the knowledge graph, forming important nodes. These nodes are crucial as they help establish the relationships and connections between different false statements.

```
def insert_statement_entities(self, row, entities):
        entity_keys = list(entities.keys())
        for key in entity_keys:
            if key == 'PERSON':
```

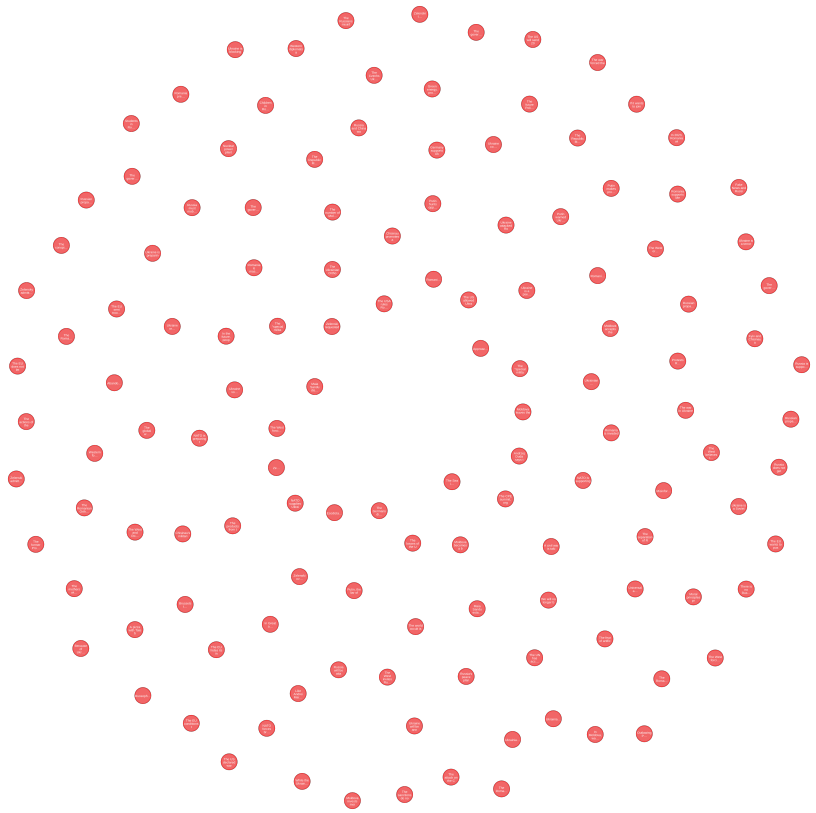


Figure 5.1: Initial graph subsection after fake statements node insertion

```
            self.__insert_person_entities(row, entities[key])
        else:
            self.insert_simple_entity(row, entities, key)
```

There are two primary types of nodes that we consider significant in our knowledge graph: Entities and Persons. Entities can be a wide variety of subjects, including organizations, locations, events, or any other significant factor within a false statement. The type of the node remains simple, as Entity, but the type of the edges varies according to the type of entity. It has the following different labels in order for further queries to be able to find patterns based on locations, time, etc. The following types of labels are used:

'MENTIONS_CARDINAL', 'MENTIONS_DATE', 'MENTIONS_EVENT', 'MENTIONS_BUILDING', 'MENTIONS_GEO_ENTITIES', 'MENTIONS_LANGUAGE', 'MENTIONS_DOCUMENTS', 'MENTIONS_LOCATION', 'MENTIONS_MONEY', 'MENTIONS_AFFILIATION', 'MENTIONS_ORDINAL', 'MENTIONS_ORGANIZATION', 'MENTIONS_PERCENT', 'MENTIONS_PERSON', 'MENTIONS_PRODUCT', 'MENTIONS_QUANTITY', 'MENTIONS_TIME', 'MENTIONS_ART'.

```
QUERY_ENTITIES = """USE news
        UNWIND $inputs AS doc
        UNWIND $ent AS ents
```

```
MATCH (n:Fake_Statement {id: doc.id})
 FOREACH(entity IN $ent |
    MERGE (e:Entity {name: entity})
    MERGE (n)-[:""" + self.dic_key[key] + """]->(e)
)"""
```

On the other hand, 'Persons' are specifically human individuals who are implicated in the fake narrative. Given the critical role individuals often play in the creation, dissemination, or subject matter of fake news, a separate category for 'Persons' is maintained. Tracking individuals as separate nodes allows us to examine the patterns of how certain persons are linked to or affected by the spread of fake news.

5.4 Entity disambiguation

When the entity in focus is a person, the software enters a phase known as entity disambiguation. This step is necessary due to the potential variations in how a single individual might be referenced in different contexts. For instance, an individual like Vladimir Putin could be referred to in various ways such as 'Vladimir Putin', 'Putin', 'Vladimir Putin, President of Russia', or 'President Vladimir Putin', among others.

Each of these variations presents a challenge for the software as it needs to determine whether they all refer to the same individual. In our approach, we utilize Wikidata as a knowledge base for this purpose. Wikidata, with its extensive collection of structured data, helps in distinguishing and correlating the various references to the same entity. It facilitates a systematic and effective process to ascertain that diverse references are, in fact, pointing to the same individual.

The software initially attempts to perform an exact match search on Wikidata. It uses Wikidata querry API and SPARQL query language to select the entity with the given label which is *Instance of* (wdt:P31) *Human* (wd:Q5):

```
SELECT ?entity ?entityLabel ?id WHERE
{?entity rdfs:label "' + str(entity) + '"@en.
?entity wdt:P31 wd:Q5.
SERVICE wikibase:label {
    bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
}}
```

If this search fails to identify a person with the given name, it indicates one of two possibilities: either the individual is not sufficiently notable to warrant an entry on

Wikidata, or the name provided is incomplete or is not in an official or recognized format.

To address the latter situation, the software implements the MediaWiki API. It leverages the API to search for the term in question.

```
SELECT * WHERE {
            SERVICE wikibase:mwapi {
                bd:serviceParam wikibase:endpoint "www.wikidata.org";
                                wikibase:api "EntitySearch";
                                mwapi:search " + entity + ";
                                mwapi:language "en".
                ?item wikibase:apiOutputItem mwapi:item.
                ?num wikibase:apiOrdinal true.
            }
            ?item wdt:P31 wd:Q5
          } ORDER BY ASC(?num) LIMIT 20
```

Once located, the term is then incorporated into the knowledge graph. This approach ensures the graph's integrity accommodating variations and irregularities in naming conventions.

```
def __insert_person_entities(self, row, entities):
    for entity in entities:
        qstr = self.__get_label_query(entity)
        response = requests.get(self.wikidata_query + qstr,
                                params={'format': "json"})
        if response.status_code == 200:
            try:
                Try exact matching of the name using wikidata
                Insert Node
            except:
                Use MediaWiki search engine
                Insert Node
        else:
            logging.error('This API request failed for entity %s', entity)
```

Upon the successful integration of entities and keywords into the system, the knowledge graph has been populated with a total of 524 nodes and 894 edges. This connected knowledge graph is now ready for exploration using a variety of algorithms. It encapsulates the intricate web of false statements, delineating how diverse claims can converge towards the same misinformation. This graphical representation therefore facilitates an in-depth analysis of the propagation and intersection of falsified narratives.

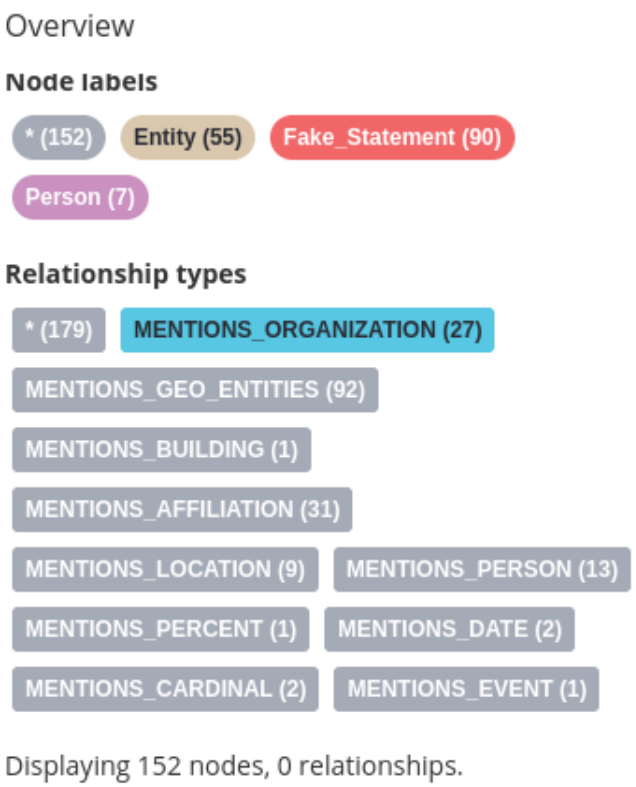


Figure 5.2: Graph overview

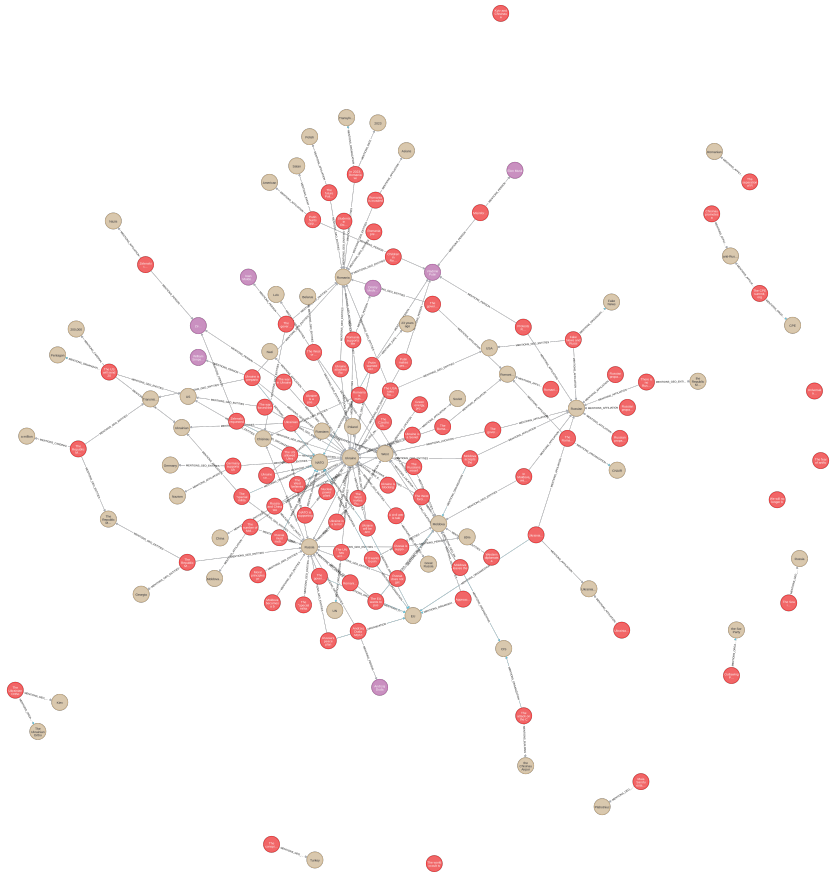


Figure 5.3: Graph subsection after entities and persons connection

Chapter 6

Search engine

In knowledge bases, knowledge graphs are frequently used to represent entities and their relationships. In contrast to relational data, knowledge graphs in the actual world lack a well-defined schema and typing system. Various query processing techniques are proposed for searching knowledge graphs. Without prior knowledge of the underlying data graph, however, it is difficult for end-users to formulate precise queries that will lead to meaningful results. Due to ambiguity in queries, inherent computational complexity, and resource constraints for large knowledge graphs, querying such graphs is difficult [SWL+18].

The MindBugs Discovery tool, described in this document, provides users with the capability to enter data using natural language. After doing an analysis of the data using its intrinsic keywords and entities, the algorithms proceed to conduct a comprehensive search for existing fact-checking activities that are in line with the provided material. This system makes significant steps towards reducing redundant journalistic efforts, therefore acknowledging and appreciating the previous contributions made by other fact-checking specialists.

The operational architecture of this system exhibits resemblances to a recommendation engine, with components that evoke standard recommendation item sets. The system provides a comprehensive analysis of the associations between the recommended entities and the initially entered information. The objective of implementing such transparency is to establish a strong correlation between the system and a white-box model, providing clarity and comprehensibility.

In the initial phase, the system performs community detection on the existing dataset, utilizing the Louvain algorithm to categorize fake statements into tightly interlinked communities. This operation is conducted on a daily basis to ensure up-to-date clustering of the database. Upon the introduction of a new statement into the system, relevant keywords and entities are extracted. Then, the FastRP algorithm is deployed to discern statements within the database that have the highest similarity to the newly entered statement. This is followed by the application of the k-Nearest

Neighbors (k-NN) algorithm, which generates a ranked list of statements exhibiting the closest similarity to the new entry. Given the system's focus on exposing the community structures, a weighted sum calculation is applied to the recommendations. This facilitates the identification of a community that is most congruent with the characteristics of the newly added data.

6.1 Louvain algorithm for community detection

The MindBugs Discovery tool has incorporated the Louvain algorithm as part of its functionality, executing it daily across the entirety of its data set. This daily analysis seeks out and identifies communities within the graph, emphasizing nodes that show strong similarities and connections. By clustering the most interconnected nodes, it becomes easier to discern patterns and relationships in the data.

These identified communities primarily consist of interconnected fake statements. Their interconnection suggests that these statements are not isolated misinformations but share common themes or sources, indicating their relatedness. This association helps in understanding the nature of the misinformation and also in deciphering the underlying patterns that might be driving them.

The algorithm's ability to quickly cluster related statements means that journalists can swiftly access insights and knowledge previously documented by their peers. This not only streamlines their investigative process but also enhances the depth and accuracy of their reporting by providing a broader context of information.

6.1.1 Louvain algorithm

The identification of communities has emerged as a basic procedure in several applications involving graph theory. This method is used to identify strong connected communities inside real-world networks, without placing any predetermined limitations on the size or number of communities. The Louvain method is an iterative heuristic algorithm utilized for the purpose of optimizing modularity.

The approach, which was initially created in 2008, has gained significant popularity due to its effective and rapid detection of community partitions with high modularity. The topic of interest is community detection. The task of community discovery, denoted as *G(V, E, w)*, involves finding a partitioning P of communities that optimizes the modularity. [LHK15]

The approach is characterized by an iterative process comprising of two primary phases that are repeated until the maximum modularity is attained.

In the local phase, the allocation of nodes to communities is performed in a way that optimizes the gain in local modularity. At the outset, every node is considered an independent community. Subsequently, the aforementioned technique assesses the modularity gain associated with the removal of each node from its existing community and its subsequent placement inside the community of its neighboring nodes. The decision to proceed is contingent upon the presence of a benefit.

During the aggregation phase, the communities identified in the local phase are combined to form a novel network of communities. The weights of the connections between the newly introduced nodes, which represent communities, are determined by calculating the total weights of the connections between the nodes inside each individual community.

Modularity Q for a particular partitioning is defined as:

$$Q = \frac{1}{2m} \sum (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \tag{6.1}$$

Where:

A_{ij} is the weight of the link between node ii and jj.

$k_{ik} and k_{jk}$ are the sum of the weights of the links attached to nodes ii and jj, respectively.

mm is the sum of the weights of all the links in the network.

$\delta(c_i, c_j)$ is 1 if nodes ii and jj are in the same community and 0 otherwise.

6.1.2 Louvain implementation

The algorithm in question is integrated and executed within the Neo4j graph database system. By running the algorithm directly on Neo4j, it operates on the data stored in the knowledge graph database in real time. This ensures optimal performance and accurate results, as Neo4j is specifically designed to handle and query graph-based data structures.

At first, a projection is created using the nodes of types Fake_Statement, Entity and Person, specifying that the relation is undirected and there is no weight on the edges.

```
CALL gds.graph.project(
    'myGraph',
    ['Fake_Statement', 'Entity', 'Person'],
    {
        HAS_KEYWORD: {
            orientation: 'UNDIRECTED'
        }
    }
```

```
)
```

After the projection is made the Louvain algorithm runs on the projection and then writes back on the graph the community property of each node. In this code, the keywords are also part of different communities, not only the fake statements.

```
CALL gds.louvain.write('myGraph', { writeProperty: 'community' })
YIELD communityCount, modularity, modularities
```

6.2 Fast Random Projection

Fast Random Projection (FastRP) is a node embedding method that belongs to the category of random projection algorithms. Their purpose is to find similarity between nodes, and their speed is often used for product recommendation system. In the presented work, the existing nodes are clustered already into communities. When a new statement is introduced by a user, the algorithms adds the node with it's keywords in the graph. It then uses Fast Random Projection to compute the similarity of the new node with the other nodes in the graph. This computed similarity is further used by the KNN algorithm to create it's predictions.

6.2.1 FRP Algorithm

FastRP, is an efficient algorithm designed to generate distributed node representations in a graph. Random projection strategies enable the effective reduction of dimensions while retaining a significant amount of distance information. The FastRP method is designed to work on graphs, with a specific focus on maintaining similarity between nodes and their neighboring nodes. This implies that two nodes exhibiting analogous surroundings should be allocated comparable embedding vectors. On the contrary, it is imperative that two nodes that lack similarity should not be allocated comparable embedding vectors.[CST+19]

Initially, FastRP builds a node similarity matrix A^k to reflect the complex relationships among nodes. The entries of this matrix are then normalized according to the stabilization characteristics of A^k. Subsequently, an ultra-sparse random projection technique is employed on the normalized similarity matrix to derive node embeddings, according to the following equation:

$$e_n^i = avg(e_m^{i-1}) \tag{6.2}$$

where m ranges over neighbors of n and e_n^0 is the node's initial random vector. The embedding of node n, denoted as e_n, is the resultant combination of the vectors

and embeddings that have been previously defined in the procedure.

$$e_n = w_0 * normalize(r_n) + \sum w_i * normalize(e_n^i) \tag{6.3}$$

where normalize is the function that divides a vector with its L2 norm, the value of the node is w zero, and the values of iteration weights are w_i. Hence, the embedding of each node is contingent upon a neighborhood with a radius equivalent to the number of iterations. FastRP leverages higher-order interactions inside the graph, ensuring both scalability and efficiency.

6.2.2 FRP Implementation

The method under consideration is effectively integrated and run within the Neo4j graph database architecture. By executing the algorithm directly within the Neo4j framework, it performs computations on the dataset contained in the knowledge graph database in real time. The utilization of FRP in this manner leverages the inherent capabilities of Neo4j, enabling the efficient processing of data and extraction of insights from the complex linkages and connections maintained inside the knowledge graph database.

At first, a projection is made with the desired type of nodes: Person, Fake_Statement, and Entity, representing the keywords extracted from the statements and the statement itself.

```
node_projection = ["Person", "Fake_Statement", "Entity"]
relationship_projection =
{"HAS_KEYWORD":
    {"orientation": "UNDIRECTED"}
}
result = gds.graph.project.estimate(
          node_projection,
          relationship_projection
         )
```

Than the fastRP algorithm from Neo4J is runed, using the property embedding with an embedding dimension of 4 nodes.

```
 result = gds.fastRP.mutate(
    G,
    mutateProperty="embedding",
    randomSeed=42,
    embeddingDimension=4,
    iterationWeights=[0.8, 1, 1, 1],
)
```

6.3 K-Nearest Neighbors

The K-Nearest Neighbors algorithm calculates the distance between every pair of nodes in the network and establishes additional connections between each node and its k-closest neighbors. The calculation of distance is determined on the attributes of the nodes.

In the current work, upon generating node similarity metrics via the FastRP algorithm, the k-Nearest Neighbors algorithm is used to identify nodes exhibiting the highest degree of proximity to a newly introduced node. The algorithm yields a prioritized list of the most closely aligned nodes. Subsequently, a weighted sum computation is executed on this list to extract the community that most likely associates with the newly inserted node. This approach provides an optimized methodology for probable community identification.

6.3.1 KNN Algorithm

The method has as its input a network that is homogenous in nature. The connectivity of the network is not a requirement; in fact, any existing links between nodes will be disregarded, except when employing random walk sampling with the initial sample option. Novel connections are established between every node and its k closest neighbors.

The K-Nearest Neighbors method evaluates each node for comparison. The k nodes exhibiting the highest degree of similarity in terms of these qualities are referred to as the k-nearest neighbors.

The initial selection of neighbors is chosen randomly and thereafter subjected to verification and refinement over numerous rounds. The maximum number of iterations is determined by the configuration. The method has the potential to terminate prematurely if the modifications made to the neighbor lists are minimal, a condition that may be regulated.

Then the algorithm chooses the k lowest distances from the list that has been sorted. The aforementioned entities refer to the k-nearest neighbors in relation to the query location.

The topic of consideration is the comparison between majority voting and averaging.

In the context of classification, the approach of majority voting is employed, wherein the categorization of the query point is determined by selecting the class that occurs most frequently among its k-nearest neighbors [DML11].

6.3.2 KNN Implementation

The methodology being examined is efficiently integrated and executed inside the architectural framework of the Neo4j graph database. The technique is executed directly within the Neo4j framework, enabling real-time computations on the dataset stored in the knowledge graph database.

Once the embedding from FastRP have been written on the projection, the KNN algorithm from Neo4J is called. It mutates the property score, which shows how similar the new node is with the other nodes.

```
result = gds.knn.write(
    G,
    topK=2,
    nodeProperties=["embedding"],
    randomSeed=42,
    concurrency=1,
    sampleRate=1.0,
    deltaThreshold=0.0,
    writeRelationshipType="SIMILAR",
    writeProperty="score",
)
```

After obtaining the prediction of the knn algorithm, the algorithm takes the top 10 predictions and creates a weighted score of the most probable communities.

```
MATCH(p)-[r:SIMILAR]-(g)
WHERE ID(p)=123
RETURN p,r, g, r.score as similarity
ORDER BY similarity DESCENDING
```

After the KNN algorithm, the top 5 neighbors of the node are returned. Then a voting mechanism is implemented, and the community of the majority of selected neighbors is selected. This majority community is considered to be the corresponding community of the node newly introduced by the user.

Chapter 7

Chat GPT Integration

ChatGPT[Ope23] is a conversational agent based on the GPT (Generative Pre-trained Transformer) architecture, designed to generate human-like text based on the input it receives. Developed by OpenAI, the model aims to provide an interactive experience for users, offering a wide range of functionalities including answering questions, providing explanations, assisting with tasks, and engaging in general conversation. Built on state-of-the-art machine learning algorithms, ChatGPT is trained on a vast corpus of text data, enabling it to generate contextually relevant and coherent responses. Although it offers a broad spectrum of capabilities, the model is not without limitations, such as handling ambiguous queries or the inability to access real-time information. Nonetheless, its advanced natural language understanding and generation capabilities position it as a groundbreaking tool in diverse applications, from customer service automation to aid in research and data analysis.

In the presented work ChatGPT is used for generating the debunking article. Once the new statement is introduced in the system, analyzed, and assigned to a community, the language model is used to create a debunking article. After the integration of ChatGPT API, it was easy to see its limitations in regard to the journalistic area. Since the language model was trained until September 2021, it had no information about well-known events such as the war in Ukraine and other important political and social events.

In order to overcome this, the language model was fine-tuned with custom data from the dataset. From each community, the central element was extracted as training data. The model was fine-tuned on this data in order to improve its results. The data fed through the API to ChatGPT is based on the community, keywords, and entities found to meaningfully correlate the new fake information to the previous one which the system has had the time to consume.

7.0.1 API overview

The OpenAI platform facilitates API access to an array of ChatGPT models, thereby providing a robust infrastructure for model integration. It offers comprehensive code samples in both Python, utilizing the Flask framework, and Node.js, offering the potential of integration using different environments. The platform hosts a variety of models, each with its distinct performance metrics and associated pricing tiers, including but not limited to GPT-4, GPT-3.5-Turbo, Davinci-002, and Babbage. For the purpose of API interaction, user authentication, and fee management are executed through a uniquely generated API key, ensuring secure and accountable usage.

Chat models are designed to accept a certain set of messages as input and provide a message created by the model as output. While the chat style is primarily intended to facilitate multi-turn talks, it is valuable for single-turn activities that do not involve any conversational elements.

Each message belongs to a certain role which can be: system, user, or assistant. The system role sets what kind of assistant the chat must be. If no specific role is assigned like journalist or data scientist, the model will by default assume its role as a ”helpful assistant”. The user role represents the question that the system needs to respond to or the task it needs to do. The assistant role represents past answers of the chat or examples of desired types of answers provided by the calling system. Because the language model doesn't have the notion of time and continuity, it must be provided with a history of the conversation in order to be able to ”continue” a discussion. Below is exemplified a simple API call for gpt-3.5-turbo model:

```
openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[
        {
            "role": "system",
            "content": "You are a helpful assistant."
        },
        {
            "role": "user",
            "content": "Who won the world series in 2020?"
            },
        {
            "role": "assistant",
            "content": "The Los Angeles Dodgers
                        won the World Series in 2020."
```

```
            },
            {
                "role": "user",
                "content": "Where was it played?"
            }
        ]
)
```

Within the context of an API call, it is possible to provide the functions associated with gpt-3.5-turbo-0613 and gpt-4-0613 models. These models possess the capability to make informed decisions and generate a JSON object that encompasses the necessary parameters for invoking those functions. The Chat API does not directly invoke the function. Rather, it produces JSON output that may be utilized to invoke the function within the user's code.

The most recent iterations, namely gpt-3.5-turbo-0613 and gpt-4-0613, have undergone a process of refinement in order to effectively identify instances when a function invocation is necessary based on the given input. Additionally, these models have been trained to generate JSON responses that conform to the prescribed function signature.

7.1 Integrating the API

The openAI models were integrated into the present work since they represent the most advanced tools available for content generation. Inside the "KG" component a ChatGPT wrapper class was created which communicates with the API. In the overall flow of the system, after the community detection of the newly inserted statement, the debunk suggestion of the statement is generated using the OpenAI models.

The model chosen for this task is "gpt-3.5-turbo" the most advanced model available both for conversation and fine-tuning. The language model takes the user's request to debunk a statement considering its role as a "fact-checking" journalist.

```
response = openai.ChatCompletion.create(
      model="gpt-3.5-turbo",
      messages=[
          {
              "role": "system",
              "content": "You are a fact-checking journalist."
          },
          {"role": "user",
           "content": querry},
```

]
)

For testing the model there were 54 test debunks generated by the model as a test set. At first look at them, the limits of openai models were clear: because the models were trained on data up until September 2021 they lacked knowledge of important events such as the war in Ukraine. This can be seen in the vague and generic debunks generated by the system. For example:

Statement: "NATO will not defend its members in case of a conflict with Russia"

Debunk: "This claim is categorically false. It is important to clarify that NATO, the North Atlantic Treaty Organization, explicitly states in its founding treaty that an attack on one member is considered an attack on all. Furthermore, the principle of collective defense is a cornerstone of NATO's mission, as outlined in Article 5 of the Washington Treaty.

NATO has a proven track record of standing by its member states in times of need. This commitment was most notably demonstrated following the September 11, 2001 terrorist attacks on the United States, when NATO invoked Article 5 for the first time in its history. NATO allies sent troops to Afghanistan as part of the international effort to combat terrorism, demonstrating their unwavering solidarity.

Additionally, NATO conducts regular military exercises and maintains a visible and credible deterrent posture. Through the deployment of rotating forces, multinational battle groups, and enhanced forward presence, NATO has ensured a strong defensive stance in its eastern flank countries, reassuring its members against any potential aggression from Russia.

It is essential to emphasize that NATO's ultimate goal is to prevent conflicts from occurring in the first place, through dialogue, partnership, and diplomatic engagement. However, if a member state faces a direct threat or acts of aggression, NATO has both the legal obligation and the collective will to respond and defend its members.

Therefore, the claim that NATO will not defend its members in the event of a conflict with Russia is unequivocally false. NATO's commitment to collective defense remains firm, and the alliance's track record highlights its steadfast dedication to the security and well-being of its member states."

As can be seen in the example above, although the information is correct, this debunking article would have been a valid one in the context of September 2021. The model has correctly identified the most important aspects of NATO organization which relate to the statement to be debunked, but in order to create a more solid debunk it needs new data about the last events.

7.2 Fine-tuning

As previously shown the debunking articles generated by the gpt-3.5-turbo-0613 are fairly good for a generative model with data until September 2021. Nevertheless the limitations of the model are obvious, and due to these limitations could not be used by any journalist in a real work environment.

In order to overcame these limitations the model needs to be fine tuned with new data. Now the questions which emerge are the following:

1. **How can the improvement in the generated output be evaluated?**

   For a human being the limitations of the model are obvious, but how can these limitation be quantified by the system. What is the measurement for quantifying how good the system works, and by what percentage it is improved after the fine-tuning.

2. **What kind of data should be used to fine-tune the model?**

   Even though it is possible to use all available data, this approach has the following downsides: the training time is quite long, making it not feasible once large data is collected to be done on a daily basis with all the data. This implies scalability issues on the system once the application starts to be used at by a bigger number of journalists or once the number of fact-checking organization grows above a certain number.

   On the other hand the pricing of training the openai models is quite high: for fine tuning GPT-3.5 turbo the price is $0.0080 / 1K tokens, and the data has an average of 2000K tokens / debunk.

To evaluate how well the system works 54 fake statements along with their debunks were selected from the dataset. In order to quantify how well the openai model works the program takes the original debunk and compares it with the generated one. This is done using cosine similarity on term frequency-inverse document frequency on the debunks.

7.2.1 Data

To enhance the effectiveness of the generative model, a sampling strategy focusing on the community structure of fake news within the dataset was implemented. This strategy aims to capture the most representative elements of each community for model training. Specifically, the statements within each community that exhibited the highest levels of interconnectedness or node centrality were identified. This was done under the hypothesis that these statements are most likely to serve as central

hubs of misinformation that encapsulate the core themes, narratives, or motifs that define their respective communities.

The search functionality is implemented within the Neo4j database environment to leverage its high-performance capabilities in handling graph-structured data. This decision optimizes query execution speed, thereby enhancing the system's overall efficiency and responsiveness.

```
MATCH (p)
WITH DISTINCT(p.community) AS comm_id, COUNT(p) AS Total_number
MATCH (q:Fake_Statement)
WHERE q.community = comm_id
WITH comm_id, Total_number
CALL {
  WITH comm_id
  MATCH (p:Fake_Statement)-[r]-(g)
  WHERE p.community = comm_id AND g.community = comm_id
  RETURN p, COUNT(r) AS rela
  ORDER BY rela DESC
  LIMIT 1
}
WITH comm_id, Total_number, p
MATCH (q:Fake_Statement)-[r]-(g)
WHERE q.community = comm_id
RETURN comm_id, Total_number, COUNT(q) AS Fake_S, COUNT(r),
        p AS Most_Connected_Statement
```

By concentrating on these highly connected nodes, the training leverages these focal points of misinformation to more comprehensively understand the overall structure and dynamics of the misinformation network. This methodology ensures that the model is exposed to a diverse yet concentrated form of data, thereby optimizing its capacity to understand different narratives while maintaining a reduced set of data.

Chapter 8

User Interface

The MindBugs Discovery platform features a web-based interface designed to facilitate user interaction, have a high availability and enhance the overall user experience. This front-end component has been developed using CodeIgniter, a robust PHP framework. Utilizing CodeIgniter not only accelerates the development process but also offers a suite of built-in functionalities, thereby ensuring that the application remains extensible and maintainable.

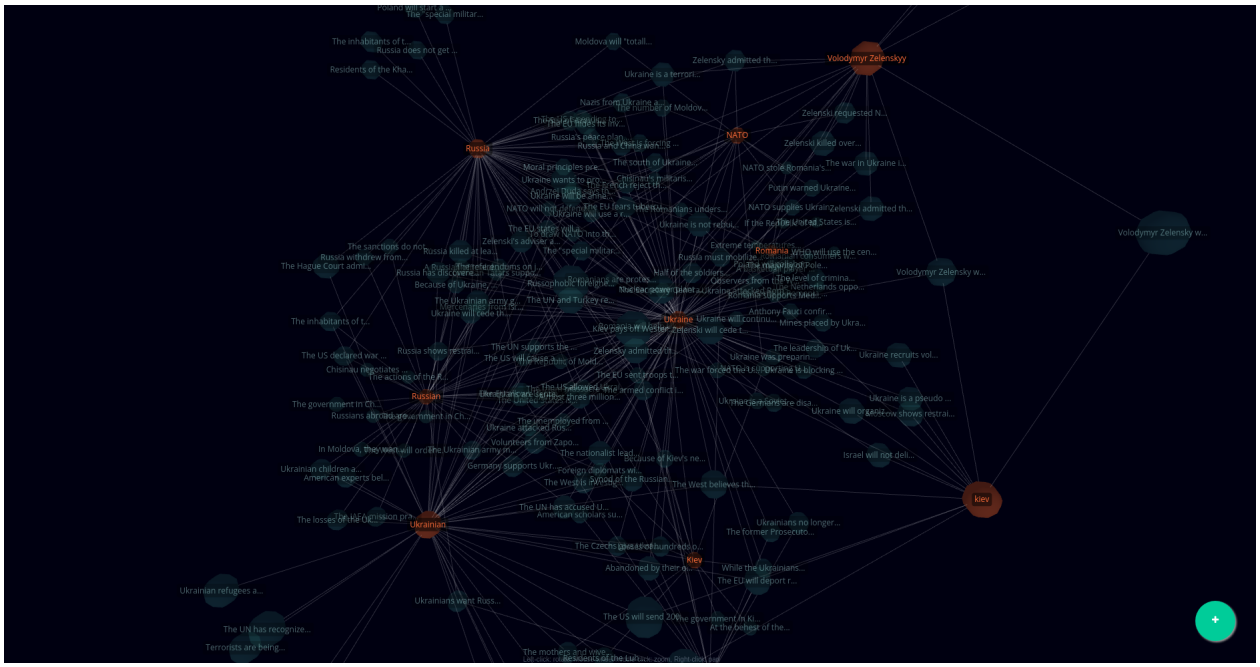


Figure 8.1: MindBugs discovery bird's-eye view

Upon accessing the application, users are presented with a curated subset of the underlying knowledge graph, sourced directly from the database. In order to streamline user experience and minimize information overload, only the top 100 most interconnected nodes are initially displayed. Fabricated statements are visually rendered in blue with lower opacity, while key information nodes are high-

lighted in a strong orange. This color scheme serves to accentuate critical focal points that are surrounded by misinformation, thereby offering a bird's-eye view upon the world of disinformation. The initial interface is presented in Figure8.1.

The visualization of the knowledge graph is created using the Force-Three-JS library[Thr23], a specialized toolkit designed for high-quality, interactive graphical representations. This library selection is chosen for its capabilities in efficiently rendering complex graph structures while maintaining high-performance standards. Utilizing Force-Three-JS enables the system to offer an immersive user experience, facilitating intuitive navigation and exploration of the graph. The library's versatility also allows for the implementation of color-coding scheme, which serves to distinguish between various types of nodes and accentuate focal points surrounded by misinformation.

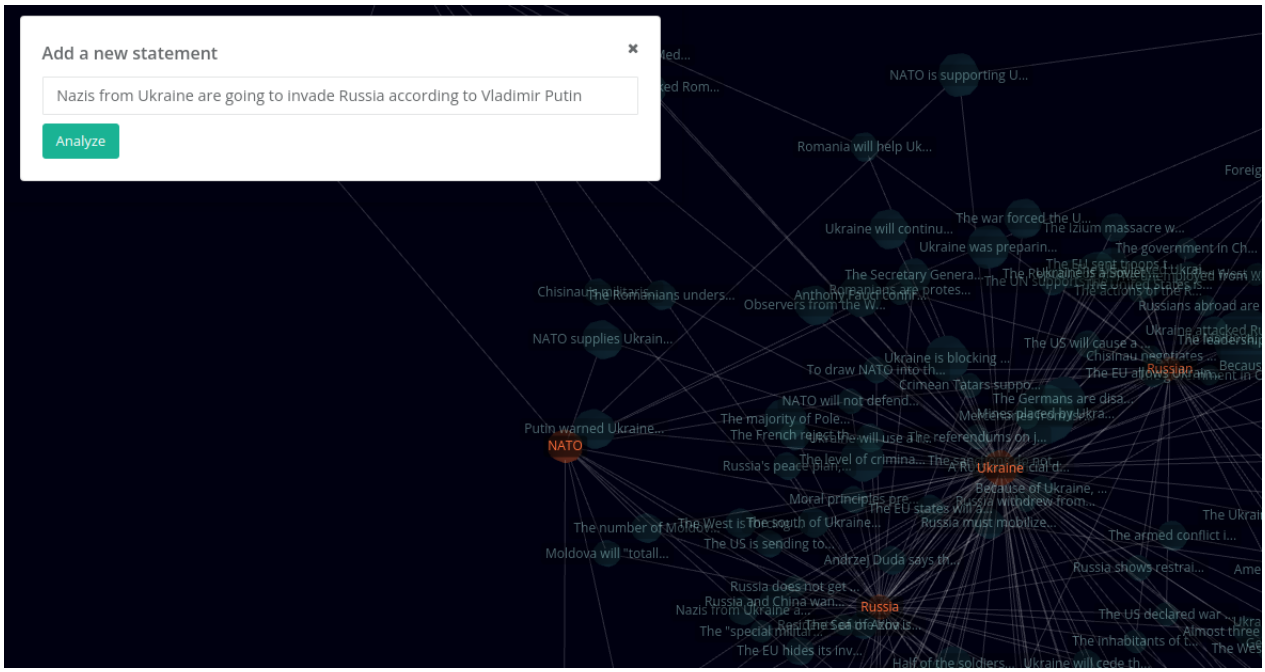


Figure 8.2: Adding the statement to be analyzed

End-users are presented with an interactive experience while navigating the knowledge graph, thanks to a suite of user-friendly controls that allow for a multi-dimensional exploration of the data landscape. Specifically, users have the ability to rotate the graph to gain alternative perspectives, zoom in and out for detailed or broad overviews, and pan across different directions for lateral examination. Additionally, the platform supports the tactile feature of node manipulation, enabling users to physically 'pull' individual nodes within the graph. This enhances not just the interactivity but also the depth of understanding, as users can intuitively explore relational dynamics among the nodes.

When users activate the displayed green 'plus' icon, a dedicated window emerges

where the users input a false statement they have encountered online or wish to investigate further. To facilitate ease of use and accelerate the user onboarding a pre-formulated statement is suggested as a starting point within the input field. The input window is depicted in Figure8.2

After pressing the ”Analyze” button the system starts to analyze the introduced data and compare it with the existing data from the system. It then displays a subgraph containing the community detected for a node. It also shows the keywords extracted from the statement and used for integration into the knowledge graph. Under the keywords extracted, it suggests different ideas and directions for the debunking argument of the journalist. The details window is presented in Figure 8.3.

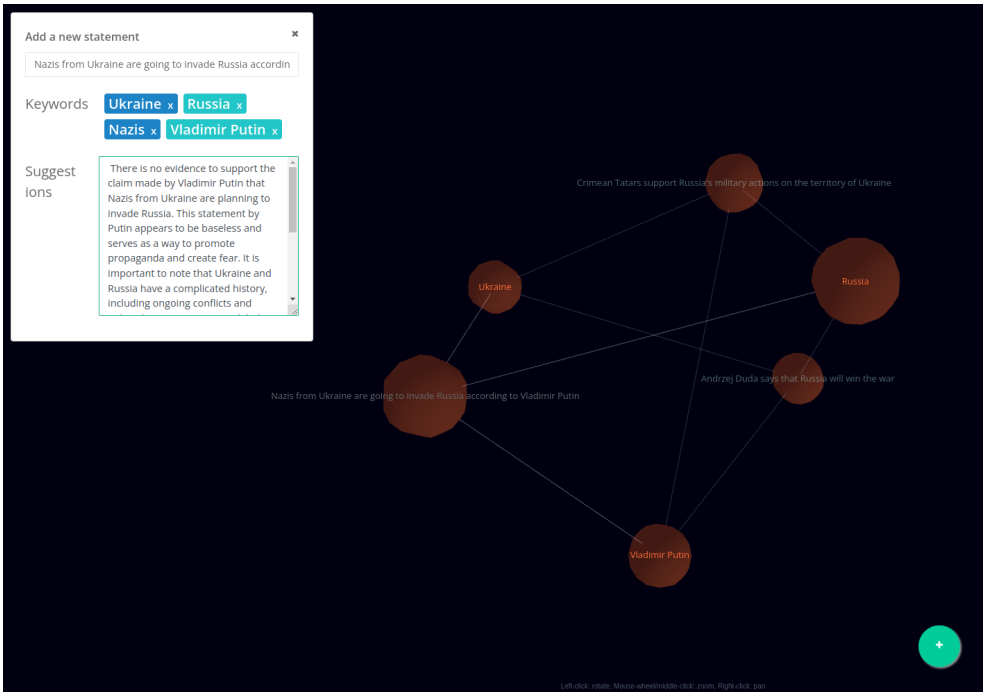


Figure 8.3: Statement analysis

As explained in the previous chapters, when the users input a statement the system processes it in order to detect similarities. It extracts information from it regarding keywords and entities involved and then connects it to the knowledge graph. Using the KNN algorithm it finds the most similar statements and then suggests the most suited community of the node. By integrating with OpenAI API a debunking article is generated which is suggested to the journalist as a starting point for his work.

The user interface tries to challenge the status quo of black-box systems. The system does not give an answer without explanation, it shows the user the links and connections which brought it to these answers. In this way, the users can decide for themselves if the criteria of correlation between the results of the search engine and their query are relevant, ethical and logical.

In conclusion, the user interface (UI) serves as the bridge between the Mind-Bugs Discovery system and its end-users. Employing state-of-the-art visualization techniques and intuitive design principles, the UI facilitates user interaction with advanced algorithms and big data. It successfully provides functionality with a user-centric design, allowing users to navigate through misinformation networks with ease and efficiency.

Chapter 9

Results

9.1 Community detection

The following results were obtained after running the Louvain algorithm on the knowledge graph obtained from Veridica dataset:

Community count: 59

The algorithm partitioned the network into 59 separate communities with the outcomes shown in Table 9.1. That means that there exist 59 clusters or groups of nodes that exhibit a higher level of interconnectedness compared to connections with nodes outside of their own communities. Out of these 59 communities, 28 have only one keyword, while the other 31 have an average of 21.1 statements.

Table 9.1: Summary of Community Analysis

Total Number of Communities	59
Communities with one statement	28
Communities with multiple statements	31
Average Number of Statements in Multi-statement Communities	21.1

A quick overview of some of the communities identified by the algorithm can be seen in Table 9.2

Modularity obtained: 0.654017706545424

Modularity is the measure of the connectedness of communities. The final modularity of the algorithm was 0,65, which is a good value. At the first iteration the modularity was 0.49, at the second 0.62 and at the third 0.652. Because on the fourth iteration the modularity increased with only 0.002, the algorithm stopped reaching a modularity score of 0.654 .

Table 9.2: Communities

Id	Nr of fake statements	Top 3 statements	Top 3 keywords
2102	65	- A Russian official demands the ”denazification” of the Baltic states, Poland, the Republic of Moldova and Kazakhstan - Because of Kiev’s neo-Nazi propaganda, Ukrainians do not understand that they are liberated from Russia, not occupied - The US allowed Ukraine to bomb Russia	Ukraine, Russian, US
111	36	- Russia shows restraint in the war in Ukraine, especially when attacking civilian targets - The disintegration of the USSR was a failure of humanity, Stalin was a pacifist visionary - Ukrainians are Russians, and Moscow has the right to intervene in the ”civil war” in Donbass	humanity, USSR, right
2130	37	- Russia killed at least 15 Ukrainian generals in Vinita with a single missile, stopping the counteroffensive in Donbass - If the Republic of Moldova had been part of Romania, it would have been an underdeveloped region, with a population of illiterate people enslaved by the Romanians - Germany supports Ukraine to forget the horrors of Nazism	Ukrainian, Ukrainians, Donbass

9.2 Search evaluation using FRP and KNN

The search system proposed starts with the Louvain algorithm for community detection and then the newly introduced nodes are searched in the knowledge graph through a combination of Fast Random Projection and K-Nearest Neighbours for their corresponding community of statements.

In order to evaluate the proposed algorithm all the data is introduced into the system. Then the communities are computed in order to find the different areas and groups of misinformation statements. From each community with more than 10 fake statement nodes, 3 random statements are removed. The nodes connected only to these statements are also removed from the knowledge graph.

The removed statements are then inserted again, with an unknown community. They are connected in the knowledge graph through the keywords and entities. The above-described algorithm tries then find their corresponding community. The accuracy score between the predicted and real communities is used as the metric for evaluating the algorithm.

The baseline for evaluating how good the community detected algorithm works is created through a simple approach in which the statement and keywords are inserted in the graph and all nodes connected to the same keywords are evaluated. The community voted by the majority of connected nodes is considered as the community predicted by the baseline algorithm.

The time for computing the communities using the Louvain algorithm is 2388 milliseconds. In case of large knowledge graphs, this time is too long for answering the query. The time needed to find the community of the node with the presented algorithm (Time_FRP + Time_kNN) is 218 milliseconds, which is a x10 improvement than recomputing the communities at the search time.

The accuracy of the algorithm is highly dependent of the FastRP parameters embedding dimension and iteration weights. The embedding dimensions represents the length of produced vectors in which the similarity between nodes is stored. The iteration weights stores the number of iterations and their impact on the resulted node embeddings. The results of running the agorithm on different data are shown in Table 9.3.

Table 9.3: Summary of FRP + KNN

Data	Parameters	Baseline Accuracy	FRP + KNN Accuracy
Keywords from both title and content	embeddingDimension=1024 iterationWeights=[0.8, 1, 1]	0.41	0.55
Title keywords and entities + content keywords	embeddingDimension=1024 iterationWeights=[0.8, 0.8, 0.8]	0.43	0.6
Title keywords and entities	embeddingDimension=256 iterationWeights=[0.8, 1, 1, 1]	0.56	0.83

9.3 OpenAI models improvement evaluation

In the presented work, OpenAI model gpt-3.5 turbo is fine-tuned on the central debunks of each community. By concentrating on these highly connected nodes, the training leverages focal points of misinformation to more comprehensively understand the overall structure and dynamics of the misinformation network. This methodology ensures that the model is exposed to a diverse yet concentrated form of data, thereby optimizing its capacity to understand different narratives while maintaining a reduced set of data.

To assess the results of the fine-tuning process a sample of 54 random fake statements, along with their corresponding debunking narratives, were chosen from the existing dataset. The program evaluates the quality of the OpenAI model's output

in comparison with the one that is fine-tuned by comparing the generated debunking content to the original. This comparison is facilitated through the application of Cosine Similarity on the Term Frequency-Inverse Document Frequency (TF-IDF) representations of both the original and generated debunks.

Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical measure used to show the importance of a word within a document relative to a collection of documents, often referred to as a corpus. In this context, each debunk represents a document in the corpus. The TF-IDF score for a specific word in a given document is the product of two separate metrics:

Term Frequency (TF): This represents the frequency of a word in a particular document. It is often normalized by dividing the number of occurrences of the word by the total number of words in the document.

Inverse Document Frequency (IDF): This measures how unique or rare a word is across the entire corpus. It is calculated as the logarithm of the total number of documents in the corpus divided by the number of documents containing the word. A high IDF score implies that the word is rare, making it more significant.

The TF-IDF representation converts each document into a vector, where each dimension corresponds to a term and its TF-IDF score in that document.

Cosine Similarity

Cosine Similarity is a metric used to determine the cosine of the angle between two non-zero vectors. It is a measure of orientation and not magnitude, making it well-suited for text comparison tasks where the length of the documents may vary but the overall orientation of the terms within the space should be similar for similar texts. The Cosine Similarity between two vectors A and B is defined as:

$$CosineSimilarity(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{9.1}$$

When applied to TF-IDF vectors representing the debunks, Cosine Similarity provides a quantitative score that ranges from -1 to 1, where a score of 1 signifies identical documents and a score of -1 implies completely dissimilar documents. This score serves as an indicator of the closeness between the original and the generated debunk, thereby offering an evaluation metric for the performance of the OpenAI model.

Fine-tune results

After the fine-tuning process was completed on the model, a marked improvement was observed in its performance, particularly in the context of Cosine Similarity scores between the generated and original debunks. Specifically, the system exhibited a 20% increase in Cosine Similarity values, indicating a more accurate and nuanced generation of debunking narratives as shown in Table 9.4. This improvement shows the efficacy of the fine-tuning process and provides empirical evidence that the modified model offers a more precise and reliable analysis compared to its previous iteration.

In order to evaluate the results I compared three models. The first model is the publicly available model gpt-3.5-turbo, the second model is trained on 56 random debunks from the dataset and the third model is trained on most connected two nodes of the communities with more than 2 statements in them.

Table 9.4: Cosine similarities

Cosine Similarity gpt-3.5-turbo	Cosine Similarity fine-tune random	Cosine Similarity fine-tune communities
Average: 0.49	Average: 0.63	Average: 0.67

More than this, when looking at the debunked arguments generated by the model, a person can easily see a more concise and updated model in terms of information. The examples below show the debunking argument generated with the standard model, with the fine-tuned model, and the debunking article written by the journalist.

Statement to be debunked: Chisinau maintains the Transnistrian conflict so that the West remains interested in Moldova

Generated debunk section without fine-tuning

There is no credible evidence to suggest that Chisinau maintains the Transnistrian conflict to keep the West interested in Moldova. This claim lacks factual basis and is nothing more than pure speculation. The Transnistrian conflict, which emerged in the early 1990s, is a complex and sensitive issue rooted in historical, ethnic, and political differences between the region of Transnistria and the rest of Moldova.

Fine-tuned debunk section

A narrative with conspiracy elements is developed by the government in Tiraspol, which accuses the authorities in Chisinau of not practically taking steps to resolve the situation in the region. Chisinau officials have repeatedly stated that they are ready to solve the situation in the region peacefully. The President of the Republic of Moldova, Maia Sandu, regularly discussed the importance of intensifying the negotiation process and finding a solution to the conflict that would satisfy both sides.

Original debunk section

It has already become a tradition for Moscow and Tiraspol to present any move by Chisinau to withdraw troops and military ammunition from the left bank of the Dniester as an attempt to abolish the peacekeeping mission, whose role is glorified and overrated. This is also mentioned by Maia Sandu in the respective interview: "[...]. Of course, it is an important task for us to find a solution, but this solution must be peaceful, diplomatic, a solution that respects the territorial integrity and sovereignty of the Republic of Moldova and the reintegration must result in a functional state. These three elements are extremely important." Maintaining the status quo, that is, the Russian military presence a little more than 100 km from its border, is neither in the interest of the EU nor NATO.

The article written by the original model is a more general one, containing some facts that relate tangentially with the subject of the debunk. Because the model lacks knowledge of the recent events it doesn't seem to find the context of the information. The fine-tuned model is more precise and puts the information much better in the context. On the other hand, it does make more "invented" facts. I could not find a metric that encompasses how much of the information is falsely fabricated, but after reading the debunks it was easy to see that some information was presented as facts while it was fabricated. The text generated with the fine-tuned model and the one written by the journalist is closely related, showcasing the improvement in the model with the new data.

The debunking argument, in its current form, serves as a preliminary foundation for investigative analysis. While it is not exhaustive and necessitates further validation through additional information, empirical facts, and research, it provides a starting point for journalists debunking the subject at hand.

Chapter 10

Conclusions and future work

In conclusion, the MindBugs Discovery platform represents a hybrid AI framework that tries to combine the power of deep learning algorithms with the transparency and interpretability inherent in white-box models. This integrated approach is particularly aimed at advancing the field of misinformation analysis and fact-checking, thereby fostering a more reliable system for addressing the challenges associated with the dissemination of false information.

The system serves as a cohesive platform that gathers the disparate but valuable efforts of various fact-checking organizations across Europe. By unifying these resources under a singular, integrated system, we not only amplify the efficacy of each individual data source but also enrich their capabilities through the mutual exchange of information and knowledge. This approach is intended to enhance the collective capacity for combating misinformation.

By architecting a modular and scalable infrastructure, the system manages to bridge the gap between data collection from credible fact-checking entities and the real-time needs of end-users, particularly journalists and researchers. The unique coupling of specialized web scrapers, RabbitMQ queuing, data aggregators, and a Neo4j graph database underpins a robust system capable of processing complex interconnected data at scale.

The system is built on data that comes from aggregated verified sources, which provides new use cases and levels of trustability by creating a balance between high volume and high-quality of data. The system further enhances its capabilities through the implementation of algorithms for community detection and similarity metrics, such as FastRP and k-NN, providing insightful context around the misinformation landscape. The successful integration and fine-tuning of the ChatGPT API for automated article debunking also demonstrates the system's adaptability to external advanced technologies and makes use of state-of-the-art technology.

While the platform shows great promise, it also has a lot of points where future research and development is needed. For instance, further fine-tuning of the lan-

guage model could make the debunking process even more accurate. Moreover, extending the array of data sources and integrating more advanced analytics could provide deeper insights into the mechanics of misinformation dissemination. Due to its micro-service architecture, the system is robust, making it easy to scale by incorporating external contributions. Some important aspects in which the system can be further improved is the addition of weights on the connections between keywords and fake statements and the improvement of community matching algorithm.

On a more general level, important steps need to be made in order to further improve the reliability, explainability and transparency of content recommendation systems. The AI of search algorithms, right now, is mostly in the form of a black-box. It gives direct and confident answers to questions or desires. These answers shape the way in which humans perceive reality. The algorithms deliver content base on characteristics hidden to users: obvious and reliable characteristics like keywords and impact or hidden ones like users' political beliefs, location or gender [TBB21].

The brain interprets the information available in order to create it's view of the world. As far as search engines are the main tool of discovering the world around, this black box approach has high ethical concerns. There is an urgent need to restructure and rediscover the way in which free choice and privacy are valued and critical thinking is developed. Critical thinking can not be supported by systems which come right away with answers, invented or not, as if there is one truth chosen according to the user's preferences. The fact that in areas like journalism or medicine, an answer without explanation is not acceptable, should not be seen as a marginal use case. Even more, considering the lack of a reliable explanation as unacceptable should be an incentive to improving the status quo. In the end, the information we feed to our brain and the way we learn to think may be as crucial as the medicine our doctor recommends.

In summary, the unique contributions of the MindBugs Discovery project are manifold. First and foremost, it gathers data from various fact-checking organizations, effectively uniting fragmented efforts into a centralized database. Secondly, the project incorporates a scalable architecture specifically designed to extract, load, and transform data, thereby ensuring system robustness and adaptability. Additionally, the system employs a novel combination of search algorithms to optimize data retrieval and analysis, thereby enhancing the depth and breadth of misinformation tracking. Moreover, the project presents a unique methodology for gathering data that is particularly beneficial for the fine-tuning of large language models like GPT through community analysis. Lastly, the system introduces an innovative approach to data presentation to the end user. Rather than merely displaying a list of elements, MindBugs Discovery leverages visualization techniques to highlight interrelationships among data points, justifying its choices in the results delivered.

This allows for a more comprehensive and intuitive understanding of the information, enhancing user engagement and encouraging critical thinking.

By harnessing a combination of advanced technologies, MindBugs Discovery meets its objectives of offering a scalable tool for journalists and also lays a robust foundation for future advancements in combating misinformation in an increasingly connected world.

Annex 1 - ”The profile: unleashing your deepfake self”

”The profile: unleashing your deepfake self” published in Multimedia Tools and Applications (2023) was the beginning of my work in the XAI domain. The article was a good starting point for what soon transformed in the MindBugs project, with all it’s different sub-projects and directions. The article can be found attached in the following pages.

Multimedia Tools and Applications (2023) 82:31839–31854
https://doi.org/10.1007/s11042-023-14568-x

The profile: unleashing your deepfake self

Ioana Cheres[1] · **Adrian Groza**[1]

Received: 26 January 2022 / Revised: 8 January 2023 / Accepted: 31 January 2023 /
Published online: 20 March 2023

Abstract

From the way we perceive society to the way we perceive ourselves, the virtual environment is changing the fundamental mechanisms of living. There is a significant gap in understanding the impact of current AI technologies on society. For instance, deepfake creates realistic forgeries that make people unable to distinguish reality from fiction, while filtering algorithms distort the perception by amplifying the interests of the user. The speed and scale of the virtual world and its hazy moral context put pressure on regulators to build the groundwork of ethical codes and regulatory frameworks. The multifaceted nature of such regulatory frameworks - legal, technological, societal, political, ethical - make it a challenging task, in which the public perception remains a key issue.

The study presents a multimedia art installation - "The Profile"- based on deepfake technologies, in which the users movements are gradually learned and an alter ego is generated. The user becomes confused when this alter ego starts to deviate from its initial mirroring behaviour. "The Profile" artwork aims to raise awareness of the main threats of current online technologies and dark patterns employed online. The users experience ethical dilemmas regarding the impact of the latest technologies in a metaphoric approach. The interaction is structured in three parts that match the steps of creating an online identity: (i) filling *personal data*, (ii) *agreeing* to terms and conditions, and (iii) receiving *personalised* content. Our work extends the current state-of-the-art by using the decomposition of the frame along with the image simplification, real time rendering and the combination of ethics and art for human-centred technology. We also discuss some of the risks of AI, along with the proposal of art as a way to approach technology ethics. With the "The Profile", we aim to raise awareness about the power of deepfakes, build new technology, understand human behaviour, and present a fresh perspective on intertwining art and AI.

Keywords AI ethics · AI and art · Full body puppetry

✉ Adrian Groza
Adrian.Groza@cs.utcluj.ro

Ioana Cheres
cheresioana@gmail.com

[1] Department of Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

1 Introduction

Today we are witnessing the modern Faustian bargain: the information people receive is almost always in accordance with their beliefs, everything else being filtered out by algorithms. GameStation has brilliantly pointed out how few customers read the terms and conditions by introducing the *"Immortal Soul Clause"*:

> *"By placing an order via this Web site on the first day of the fourth month of the year 2010 Anno Domini, you agree to grant Us a non transferable option to claim, for now and for ever more, your immortal soul. Should We wish to exercise this option, you agree to surrender your immortal soul, and any claim you may have on it, within 5 (five) working days of receiving written notification from gamesation.co.uk or one of its duly authorised minions."*

The experiment has shown that 88% of the clients did not read the clause, while only 12% clicked on the relevant opt-out box, thus obtaining a 5 GBP gift voucher (additionally to keeping their legal ownership of their soul, as a lawyer might advocate).

A complete and valid perspective, gathered from a multitude of different subjective information sources, is the ideal context in which a person can make decisions. The growth of social platforms has made them slowly become the main source of information for a large percentage of people [42]. The information users receive is being filtered by algorithms that rank each post based on likes, comments, and time spent watching in order to ensure comfort and deliver "only" desirable content. This creates well-separated and polarized communities where the users' perspective is narrowed to its beliefs, amplifying the factionalism of the modern world [6]. The factionalism of the modern world is amplified in the echo chambers, which enables social media to be controlled and guided by emerging radicalized groups [5].

The posts the viewers see are not only filtered but they can also be forged. Deepfake has become a popular technology which allows the creation of fake media on a targeted person. Creating deepfake content is so easy that anyone, with no technical skills, can use it on public websites such as Deepfakeweb [16] or MMasked [39]. Both websites have clear instructions on how to create a deepfake, with illustrative tutorials: upload a video or an image of the target person, upload the video for the target person to appear, pay and enjoy the deepfake generated video. This technology has already started to be weaponized to harass persons by creating deepfake pornography [20]. For instance, Sensity AI estimates that more than 90% of all deepfakes concern non-consensual pornography [22]. Being a victim of deepfake can be a serious problem and due to the noncentralised nature of the internet the fake content may never disappear, irreparably damaging the public image or mental health of the victim.

The distortion of information does not only affect the perception about society but also changes the opinion and standards about the self. The way humans measure their success is by comparing it with that of others [19, 36]. Seeing society through the veil of social media affects self-esteem because people compare themselves with virtual profiles. Jean Baudrillard was already anticipating these trends in the 1990, by comparing media with a genetic code that mutates the real into hyperreal. He also observed the successive phases of the image: (1) reflection of reality, (2) masking or denaturing the reality, (3) masking the absence of reality and (4) being its own simulucrum with no relation to reality whatsoever: a hyperreal [2].

AI algorithms are quick to remove imperfections, improve complexion features, or modify body proportions, thus offering a better version of themselves. There is no surprise that

photo editing behaviour is very popular [34] making people feel as if everyone around had the perfect look. This is seen as a cause of increased dissatisfaction of people with their own body [36] and is affecting persons of all ages [6]. Because of the unrealistic standards, which are deeply embedded in people's perceptions, social media users with heavy photo-editing behaviours are found to often suffer negative psychological consequences like body dissatisfaction, low self-esteem and even leading to eating disorders [34].

A comprehensive evaluation of twenty peer-reviewed scientific studies found evidence that the used of social networking sites is related with greater body dissatisfaction and eating disorders [27]. The indicators of body dissatisfaction, eating disorders, increased endorsement of the slim ideal and aesthetic comparison were found to be directly related to the time spent on social media throughout the chosen studies. Because social media encourages active usage over passive consumption, some research attempted to assess body dissatisfaction in relation to the user's involvement time. This metric was directly related to objectified body consciousness, which predicted greater body shame [37]. According to sociocultural and objectification theories, these findings [27] show that spending more time on social media is associated with female body dissatisfaction, at least in part owing to the internalization of unrealistic ideals, self-objectification, appearance comparison and self-surveillance.

The AI advancements impact society as a whole, not only the specialists who understand and develop these technologies. Our society needs a way of presenting and analysing meaningful ethical problems on a large scale. We argue that the latest forms of art created using AI technology and promoted by organizations like ArsElectronica (https://ars.electronica.art) or NestaItalia (https://www.nestaitalia.org) are means to involve the population in the analysis of complex ethical issues raised by technology advancements. As stated in About Us section of the ArsElectronica webpage: "Together with artists, scientists, technologists, designers, developers, entrepreneurs and activists from all over the world, we address the central questions of our future. The focus is on new technologies and how they change the way we live and work together", these organizations create collaborations between people concerned with this problem. The resulting art projects address ethical questions regarding the future of humanity and bring research data and experiments close to the general public. By presenting information in a graphical and interactive way, the public has the chance to develop an intuitive understanding of the main threats and opportunities the latest technology offers. The exhibitions are open to people of all ages, making not just adults realize the impact of technology on society, but also giving to children an early sense of how technology works. Organisations such as ArsElectronica and NestaItalia are about much more, but in the current context the ethical issues of technology are a relevant component of their mission.

Along similar lines, we have built an interactive art installation called "The profile" [29]. In "The profile" users experience ethical dilemmas regarding the impact of the latest technologies in a metaphorical approach. The interaction is structured in three parts that match the steps of creating an online identity: (i) filling in *personal data*, (ii) *agreeing* to terms and conditions, and (iii) receiving *personalised* content. With the "The Profile" we aim to raise awareness on the power of deepfakes, and present a fresh perspective on intertwining art and AI.

At first, the users explore a virtual environment by projecting ideas in the physical world using their own bodies. Each user is becoming an active cog in the mechanisms of spreading fake, aggressive and erotic content, being faced with some of the major social media trends

that influence society. The content is structured in three circles of hell inspired by Dante's Divine Comedy: limbo, lust and wrath.

Second, the user agrees to the terms and conditions, allowing the software to use private data and narrow the users' perspective. From this point on, the users will see only images that mirror their appearance, which are mimicking the delivery of filtered content according to users' preference of any type of social media that has the content delivered by AI algorithms. The process can be seen as the modern Faustian bargain, which is essentially the renunciation of any form of self-determination, decision, free will and the transfer of decision-making authority to an external entity. It is a convenient position, in which the truths are clear and come ready, nullifying the need for discernment.

Third, the users are confronted with a deepfake of their own body. They can see themselves as in a mirror, but the software takes control of the image at random intervals, increasing the confusion between what is real and what is generated. The users experience first hand the impact of private data manipulation: they are, for a short period of time, victims of deepfake.

In the remainder of the article we present the technical challenges of building this artwork. Section 2 browses related work on deep fake and other AI-based art installations. Section 3 specifies the hardware and software architecture of the installation. Section 4 details the experiments and results, while Section 5 deals with the artistic and psychological dimension of the project.

2 Related work

Generating artificial images with neural networks remains a challenging task. Based on the input of artificial network, there are three main approaches to generating fake images: (i) using a random vector (i.e. noise) [24], (ii) a text description [35], or (iii) another image [30]. The base architecture for neural network image generation are generative adversarial networks (GAN) [24]. The GAN architecture contains two neural networks called the generator and the discriminator that play a minmax game. The generator takes as input a noise vector and has to output an image, while the discriminator takes as input an image and has to output the probability of that generated of being a forgery.

The structure of the GAN can be modeled according to the needs. More discriminators or generators can be added, each specialised in a different task. In case of generating story visualization, an approach is to use two discriminators, one for detecting story inconsistency and one assessing the realism of the generated images [35]. Another approach is to use a markovian discriminator that classifies each $N \times N$ patch of the image and outputs the final result as an average of all the patches [30, 47].

The input data shape and content drastically influences the performance of the neural network. The input data represented by a text can be changed into a multidimensional space using a story encoder [35] or it can be another observable image [30].

A popular motion transfer approach is the work of Chan et al. called "Everybody Dance Now" [11], in which a fake video of users is generated where they are dancing like a professional dancer starting from the library OpenPose [8] which is a collection of functions for skeleton representation. For the realism of the images "Everybody dance now" [11] project uses separate GANs for face and body in order to maximize the human aspect of recognizing identity: that is increasing the realism of the face.

Recent work in video to video synthesis [46] offers image translation based on a general abstraction of human images obtained from sketch filters on the original subject. To

create videos that automatically generate content of a person mouthing words using existing footage has been done with Video Rewrite [7] by mapping facial keypoints between the subjects. The project focuses on facial expressions and mouth position and not on full body position transfer. MoCoGAN [44] uses MUG Facial Expression Database [1] to map expressions on target faces and extends this capability to full body image retargeting using a noise vector as input.

From an artistic perspective, the AI is compared by Aguuera [4] with the invention of applied pigment. Algorithmic art, which is a term used to describe any artwork that cannot be created without the use of programming, has captured world wide headlines and sells at auction at considerable prices [41]. Important art exhibitions like Athens Biennial[1] host artificial intelligence artworks like "Seamless" by Theo Triantafyllidis, and there are even art museums and exhibitions dedicated only to AI [41].

The AICAN project studies the artistic creative process and the evolution from perception to cognitive opinions in the context of art generated by programs [38]. It uses a Creative Adversarial Network (CAN), that is a variation of GAN with stylistic ambiguity used for improving the complexity and novelty of the resulting images [18]. Hong and Curran have found that people could not distinguish CAN-generated artwork from human-created artwork, and that CAN-based art receives better scores in terms of novelty, visual structure, or inspiration [28]. More than that, Google has launched Deep Dream Generator, a set of AI tools for creative visual content generation.

It has been argued by Chomanski [13] that ethical codes of big tech companies are nothing more than covers for the big tech commercial practices and a strategy for avoiding regulation by law, hence a form of "ethics-washing". The aim of transparency and explainability through regulation is to balance the information asymmetry between developers of AI applications and users of AI. This explanatory trend is also reflected at the technological level: starting from the *expanding phase* of neural architectures (more layers, bigger and bigger data), continuing with the *diet phase* (e.g., few shot learning, knowledge distillation, ablation studies), and the current *explanatory phase* (XAI, natural language understanding).

3 The profile architecture

The rising of AI in visual art has opened a new world of possibilities in exploring and creating meaningful and complex artworks. "The Profile" installation is based on recent advances in the realistic image-to-image translation, pose estimators and art experiments. Modern pose detection, (based on the OpenPose [8]), and image-to-image translation (based on *pix2pix* neural networks [30]) are the technologies on top of which "The Profile" is built [29]. The art installation has four components: (i) physical devices and their relative location, (ii) the distributed architecture, (iii) the local component for user interaction, and (iv) the remote component for heavy computation.

3.1 The physical components

The location of the physical devices is relevant for the quality of the training images. Since it is an interactive application, the images are taken while the user is using the installation

[1] https://athensbiennale.org/en

and there is a limited amount of time to process them. That makes it a computational challenging task, so the position of the elements ensures a simplified image with no background and shadows. During the experiments, we set a time limit of 12 minutes, out of which the generation of the fake body requires 9 minutes. The physical components of the installation are: the webcam, the screen, the background and two lights, positioned as in Fig. 1. The webcam is the main source of information for the system that captures frames of the subject. The background needs to be in one colour for the program to quickly crop the person out of the frames. There is an important relationship between the two light sources: light 2, that is between the user and the background is set three times stronger than light 1, which is between the user and the screen. This placement of the lights prevents the user from casting any shadows on the background, which could complicate the structure of the frames. It also adds a frontal light on the subject for diminishing the shadows on the body, while allowing details like the texture of the clothes and skin to be captured accurately on the frames.

"The Profile" can be experienced during MindBugs' AI&Art exhibition in Cluj-Napoca, where visitors discover the functionality of the artwork out of curiosity. Due to the nonconformist manner of conveying a message, the viewers have to draw conclusion and meaning from the experience, without external influences. An art exhibition is the appropriate space where the viewers can experience different stimuli and search for meaning in what they see. Before entering the installation, the users are informed that their data will be processed to create a deepfake. They are also notified that some of the images might have an emotional impact and that after the experience all their data will be erased. They are asked for their permission and are explained the process.

On the screen the users see their skeleton, which they control by moving their body. Since the generation of the fake bodies requires 9 minutes on average, we introduced some gamification elements in form of 3 levels or tasks of about 4 minutes each. In each level

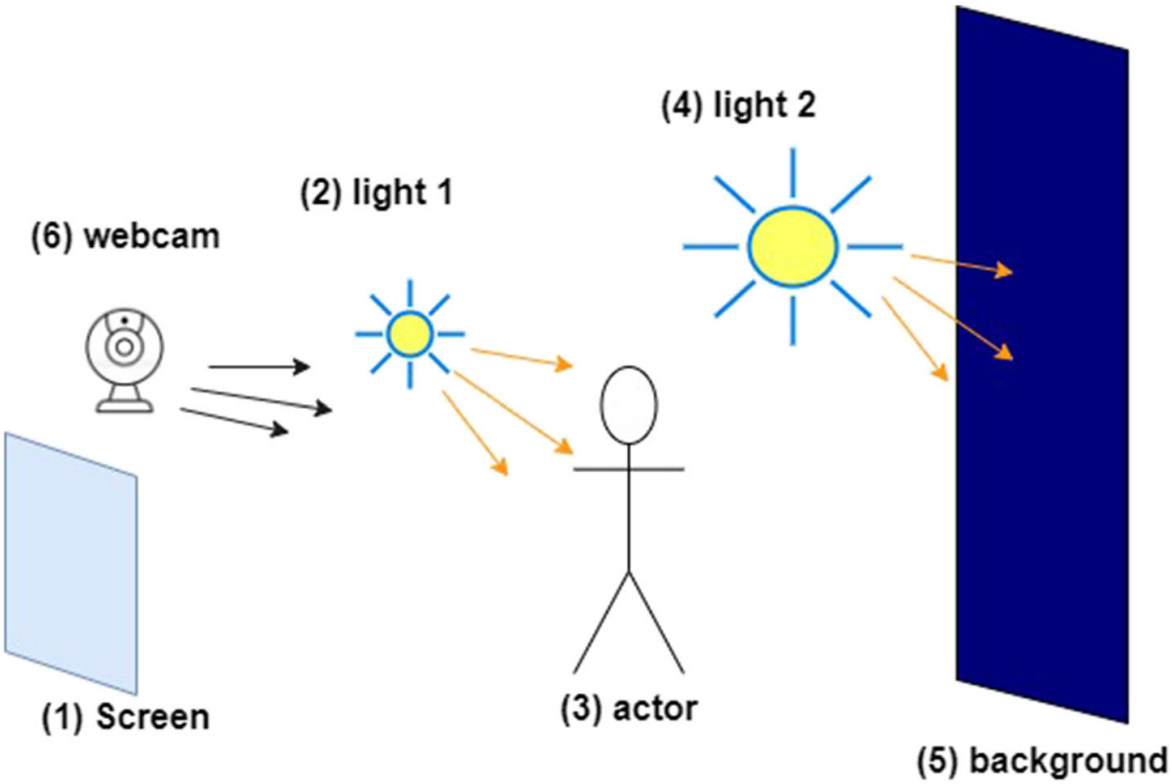


Fig. 1 Physical components of the installation ©Ioana Cheres, 2021

the users have to overlap their bodies against some visual content displayed on the screen. When the users succeed to pose their skeleton in the required positions, some hidden images and messages are revealed. These images and messages are related to the theme of the art installation: awareness about the power of deepfakes. These gamification elements guide the interaction, aiming to be able to generate the fake bodies.

3.2 The local software component

The local component parses each frame and computes the image displayed on the screen. In the first two circles, the users explore the platform and interact with the environment using their skeleton, which represent their virtual ego.

Each frame taken by the webcam is decomposed in three separate images as shown in Fig. 2: skeleton, background and person. The skeleton is generated using the OpenPose library [8] and has as input the current frame. The background is already known and, due to the position and intensity of the lights, it can be extracted from the frame to obtain the person only. Once all three secondary images are computed (i.e. the skeleton, the background, and the person), the skeleton and person images are saved for training the generative adversarial neural network.

The Scene Composer component takes all the partial images computed and creates the frame to be displayed on the screen. From the person image a mask is computed, that acts as a display area for the scene components, giving the illusion of projecting through the body. The part of the body that is not overlapping any components is replaced by the skeleton, and the ring of hell is added behind the visible components and the skeleton. The ring of hell is a visual representation of sins inspired by Dante's [3] book. After the main elements are computed, the background image is added. All the elements are treated as layers and the operations on final frame composition are done using the OpenCV collection of functions [40].

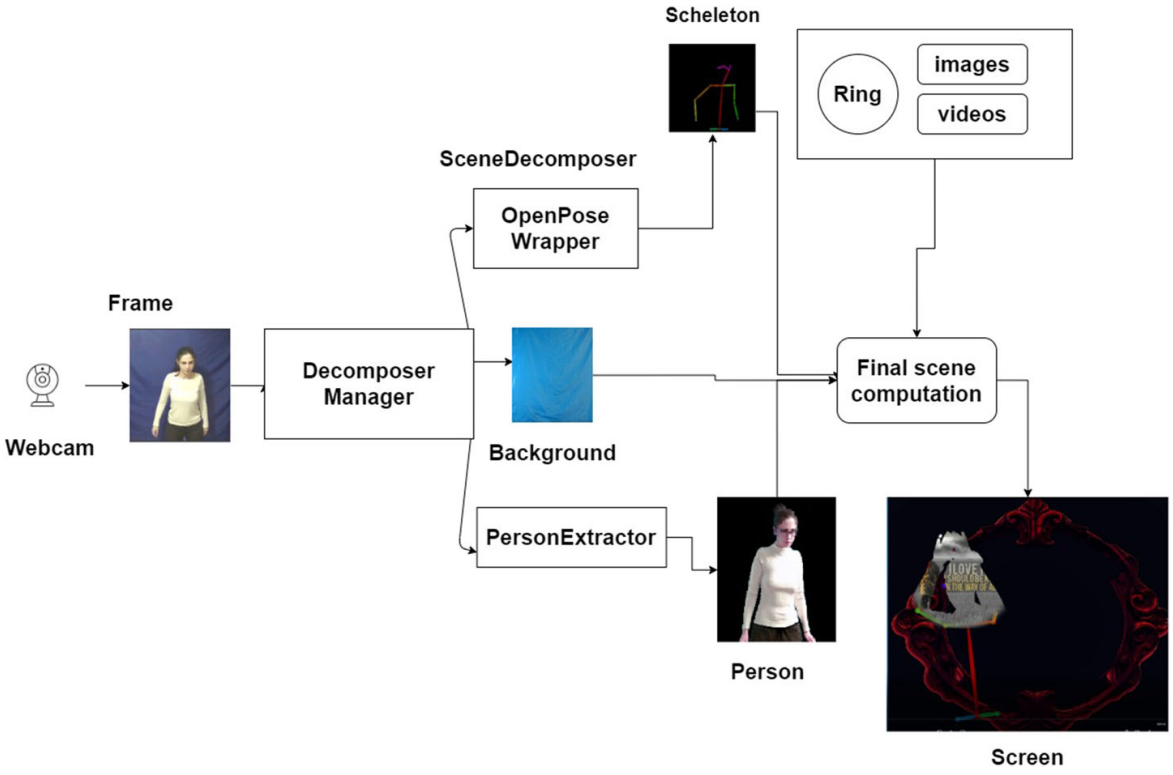


Fig. 2 Decomposing the scene into skeleton, background, and user ©Ioana Cheres, 2021

When the subjects have to pass from one level to another, they have to place their hands inside two circles. The Euclidian distance between the center of the circle and hands is computed and, if it is smaller than a value ϵ, the user passes to the next level.

While the user explores the first two parts, the system continuously saves pairs of images of the person and the matching skeleton and sends them to the remote component for the neural network computation. When the user finishes exploring the first two parts of the installation, the system downloads the learned model.

In the third part the system captures the images of the person and computes the skeleton, and then it generates the fake body. The users can see themselves on the screen, in the exact position they are standing, with the bodies generated instead by a neural network. At random time intervals, the system takes control over the image of the user. It alters the joint position of the skeleton and feeds forward the neural network with the computed skeleton, resulting in an image of the subject in a different position than it currently has. The program has specific algorithms that modify the joints in order for the image from the screen to wave to the subject, dance and move from side to side. To increase the psychological impact, the system computes the sequence of independent moves to start and end in the current position of the user.

3.3 The remote software component

The remote software component is responsible for heavy computation. It has the objective to train as fast as possible a deep neural network that can predict the body starting from a skeleton image. The network used is a *pix2pix* [30] generative adversarial neural network. A *pix2pix* neural network learns the mapping from the skeleton to the body image. It is composed of two smaller neural networks called the Generator and the Discriminator.

The Generator learns to compute realistic images of the body, having as input the skeleton from Openpose. The Discriminator learns to predict which images are real and which are computed by the Generator, as showed in Fig. 3.

The structure of the GAN follows the *pix2pix* [30] architecture, in which an artificial neural network learns the mapping between two images called the source and the target. The Generator has eight layers of encoding, seven layers of decoding and skip connections. The Discriminator is a PatchGAN that classifies a 70×70 patch of the input image. The

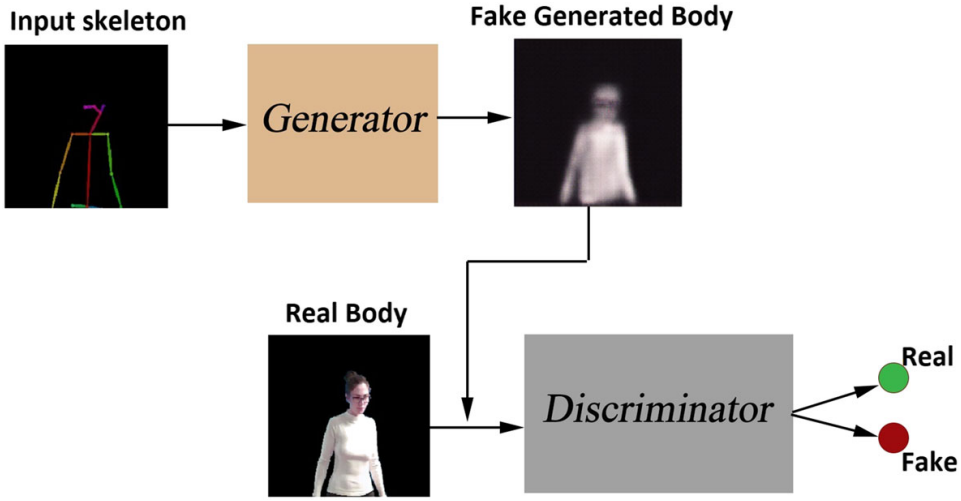


Fig. 3 Applying GANs to generate the fake body ©Ioana Cheres, 2021

data is resized to match the size of the input layer, random flipped in order to increase the generalisation capabilities and then normalized.

The decomposition of the frame, along with the image simplification, real time rendering and the ethical dimension of the project, are the most important components that extend the current state of the art and offer a new perspective in this field.

3.4 The distributed architecture

In the first two parts the users interact with the platform which captures images of them. In the third part, the users see on the screen a deepfake of their own body. Due to the complex capabilities needed, the system has to be distributed from a hardware perspective. The system needs to be moved easily between art exhibition and it should function in various environments (e.g. dust, light), but it also has to have a high computing power in order to learn the model in less than twelve minutes. Hence, the main component is at the location of the user and is a part of the installation. It has the purpose of interacting with the subject in all three parts and capturing the data. The local hardware has reduced capabilities and performance, but can be moved easily and is more robust and enduring to external factors.

When enough images of the subject are taken - that is [300-375] images in our experiments - they are sent to the remote component to train the GAN for generating the fake body. The remote component is DGX machine with nine TeslaV100 GPU and 700GB RAM. The local component runs on Windows and has a GTX 1660, 6GB RAM and i9 processor. The webcam has a resolution of 1920×1080 pixels and 30 frames per second. The main component uses the Paramiko library [48] that implements the Secure Shell (SSH) protocol to securely send and receive data. It sends Linux commands over SSH and starts on the remote component an Nvidia docker container [31], which has the Python code that trains the neural network.

4 Experiments and results

To test the performance of the GAN, the system captured 745 pairs of skeleton and body images. For evaluating the GAN, two metrics are used: Mean square Error (MSE) and Structure Similarity Index (SSIM) [11, 30]. Figure 4 shows how the two metrics evolved in 100 epochs, which are 100 cycles of training. One can see that the neural network learns in 20 epochs to generate a realistic body of the subject.

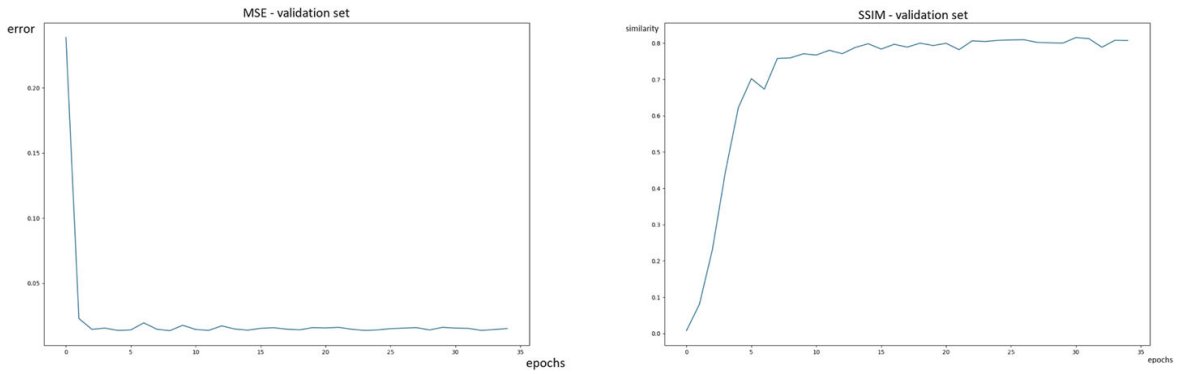


Fig. 4 Generator evaluation ©Ioana Cheres, 2021

The MSE is decreasing abruptly and then constantly following a straight line, meaning that on pixel to pixel differences the images have a high quality. When looking at the SSIM metric, the graph shows the partial instability of the generated data. On average, after 20 epochs the SSIM is larger than 0.8 which is a good result (right part of Fig. 4). SSIM is a more accurately metric when dealing with image quality because it takes in consideration the structural similarity of the images, not only pixel differences like the MSE.

Figure 5 depicts the results obtained in the first 20 epochs, which is the optimum number for training the GAN. The neural network learns fast, and even after one pass through the data, it can output the general shape of a human. As early as the 10^{th} epoch shadows start to appear, while in the 20^{th} epoch a clear distinction occurs between the texture of the clothes, skin and hair. In total, the system trains on the data in 4 minutes. The time of sending the images is 2 minutes and the time of receiving the weights (i.e. learned model) is 3 minutes. Hence, the system trains the remote component in around 9 minutes, which was one of the requirement that makes the application usable.

5 The artistic and psychological dimension

From an artistic perspective the project is a unique combination of the latest technology, with elements taken from literature. The quote from Faust [23] and the reinterpreted rings of hell from The Divine Comedy [3] are the guiding elements of the users in their experience, marking the passing from one level to another (Fig. 6). They also represent principles and ideas that the subject is already familiar with and can associate with the new trends from social media.

The approach in which the viewer is placed as a central rational mind and confronted with ethical issues is what art excels at. We have aimed here at combining senses, expressing feelings, or creating experiences that highlight the complex nature of today's ethical problems.

In the first part, the user discovers the three circles of hell inspired by Dante's work [3]. Each circle has a different chromatic scheme that matches the information presented

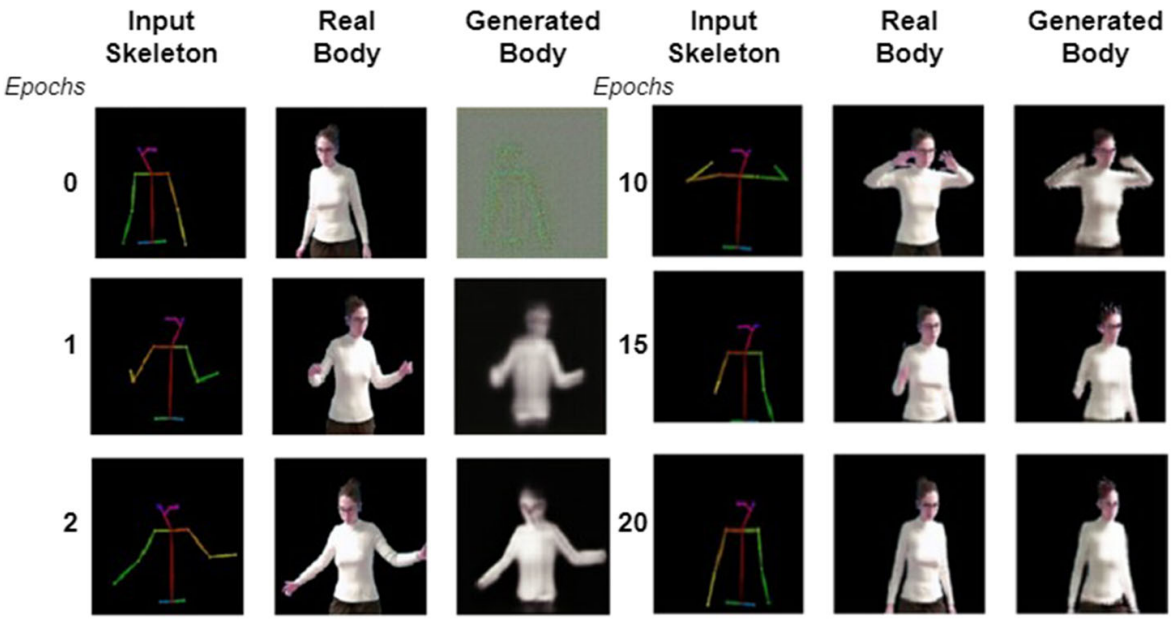


Fig. 5 After 20 epochs of training there is a clear distinction between clothes, skin and hair ©Ioana Cheres, 2021

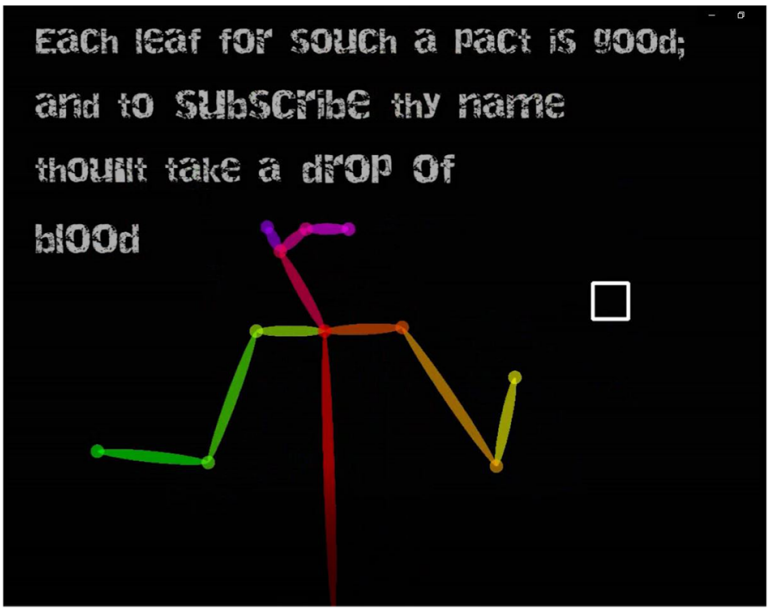


Fig. 6 Faust scene ©Ioana Cheres, 2021

and amplifies the psychological impact. The Faustian bargain has to be signed in the modern way, using a checkbox. This scene is the only one with a complete dark background (see Fig. 6), aiming to suggest the solemnity of the moment and to contrast with the other elements.

The next part focuses on the psychological impact of deepfakes. In the online era, psychopathy is positively associated with engagement in a higher frequency of trolling behaviours, due to the reduced capacity of experiencing empathy [20]. Videos with deepfakes can alter human memories and even implant false memories, as well as change people's views about the target of the deepfake. A recent study found that exposing individuals to a deepfake depicting a political person dramatically affected their sentiments against that politician. Concerningly, given social media's ability to target content to particular ideological or demographic groups, the study found that microtargeting the deepfake to groups most likely to be insulted magnified this effect when compared to sharing the deepfake with a wider public [17]. In the same line, we transform, for a short period, the subjects of the installation into victims of deepfake. We intend to give them a glimpse of the potentially harmful impact of this technology and to make them start questioning the authenticity of the contents of online videos.

By creating an AI art installation that lets non-technical users experience what AI can do, we aim to showcase the power of AI, along with training people to become more aware if AI capabilities. In today's world, there should not be anyone, tech or non tech, that is not aware of what a deepfake is and how information is distorted. Democracy is built on an educated society that is not controlled by false information, but a big part of the available information today is made of forgeries and everyone should know how easy, realistic, fast and confusing these forgeries are.

Humans tend to perceive that mass media manipulation has a greater impact on others than on themselves. This is called the third-person bias [26]. Our art assemble makes the users aware that they are the ones who are 'deepfaked'. The users face the situation of their own body puppetry. The confusion is amplified by the fact that the puppetry occurs at random intervals. The intended psychological effect is that the users should doubt their own real movements. Since the display works both as a mirror and as body puppetry, the

users encounter difficulties to figure out when their own moves are mirrored or faked. The metaphor of the mirrored-self is applied by Danesi (2018, p. 57) [15] to the entire society claiming that: "The Internet is one huge mirror metaphor, so to speak, connecting domains of all kinds [...] We might become controlled by the whole systems of bots [...] which may dupe us [...] to believe that intelligence and consciousness are independent of the body.". With the deep fake technology in their weaponry, the bots can create their fake-body to embody artificial intelligence.

The impact of deep fakes can range from individual to the society level. Specific deepfakes techniques range from face manipulation (e.g. facial expression manipulation, face morphing, face swap, face generation) to full body puppetry. Our system requires 9 minutes of training, after which real time manipulation of the body posture is enabled at random instances. The deepfake creators are now developing algorithms able to generate deepfakes based on as little input as possible: video from a single image of the target person, or speeches based on few seconds of audio [49]. For instance, Deep Angel application erases objects from photographs. The users can take the Deep Fake Test, to guess which photos are real, and which have been deep faked by AI. In the same line with Groh's Deep Angel, our art installation points out towards the "recovery of what has been lost in this Internet age: presence" [25]. While in Deep Angel the absence of the object is obvious, in our approach the absence of the real-self is gradually revealed. Once the deep-faked body takes control, the users start to realise that their presence in the social media agora is no longer genuine. The gradual renunciation to authenticity in favor of our "optimised image" has feed a new-self, that no longer represents our true-self in social interaction. Both Deep Angel (*via negativa* or absence) and The Profile (via rising awareness of the gap between the alter-online self and the real one) have the same goal: to reclaim presence of the authentic self.

Interactive GANs open opportunities for artists and designers to explore new ideas [10]. Deepfake provides artists with new tools to express their creativity, from people featuring themselves in movies [33] to visualising dreams about themselves [12], or to create memorial videos (e.g. Dali Museum [9]).

The recent proposal of the European Commission for regulating AI [14] requires, through Article 52, that the creators of deepfakes be obliged to clearly label their content. As an example, raising awareness of fake news was demonstrated on the British Chanel 4 parody, on the 2020 Christmas message of Queen Elizabeth II. In the fake video, the Queen performs a TikTok dance challenge. The fake message was clearly labelled as deepfake and broadcast at the same time as the official message [45]. However, Article 52 also states that this obligation does not hold in some exceptions, among which "the right to freedom of the arts and sciences". These attempts to regulate AI is a step towards a new generation of human rights, including epistemic rights, that are so needed in the current age of surveillance capitalism [50].

The Profile represents an interface and integration between humans and episteme-techne, showing that the capability of human-made machines is to transform user's body and mind. In our approach, the interaction between human and machine starts with the learning phase, and once the fake-body is created, the human agents lose control over their movements. The MIT project Alter Ego is more ambitious, since the interface between mind and machine occurs through *silent speech* [32]. Silent speech means that the human agent can converse with machines simply by articulating words internally. The Alter Ego project uses a face mask that interprets non-visible neuromuscular signals. That is, no voice or no movements. The objectives of The Profile and Alter Ego are also different: The Profile aims to increase awareness on the informational asymmetries between AI systems and humans, while Alter

Ego aims to help people with speech disorders. Alter Ego also aims to weave AI into human personality as a "second self" [32]. This "second self" is the main theme in The Profile installation.

The Profile installation addresses the concept of interpretability, from the philosophical perspective of digital hermeneutics, but also from the perspective of interpretability of artificial intelligence and meaning creating. From the perspective of how meaning creating is relevant or not in AI, there are two main tribes: (1) "knowledge representation and reasoning" and (2) the tribe of "machine learning". In power now is machine learning, due to a very charismatic leader: deep learning. This leader excels at solving classification tasks, mainly by "data torturing". Data torturing is executed with machine learning algorithms that query the data: "What do you hide? What do you hide?". As in any torture, the data eventually confesses its hidden statistical patterns. The problem is that if data is interrogated with "decision tree" tools, it confesses something; if data is tortured with support vector machines or artificial neural networks it confesses something else. And no one is able to figure out which confession (i.e., the learned model) is the real one. For instance, state of the art language models have billions of parameters, with no chance to extract a "meaning" from them. Since current machine learning is characterised by "epistemic opacity and theory agnostic modelling" [43], Srećković et al. have coined the term "automated Laplacian demon". This demon has the power "to turn science towards predictive and not explanatory goals" [43]. Against this demon and its failure of meaning creating, the first tribe of knowledge representation and reasoning initiated a no-confidence motion entitled Explainable AI (XAI). XAI is aiming to assure explainability, transparency, and a more human centered approach on AI [21]. The XAI program aims to enable humans to understand how AI is calculating its decisions, in order for them to be able to appropriately trust and manage the AI coworkers that they have to interract with.

6 Conclusions

"The Profile" is an AI-based art installation that uses deep fake to rising awareness to the users of the main threats of some online technologies and dark patterns running in social media. It laso rises awareness on the gap between the alter-online self and the real-one, as the online or second-self is the main theme here. The main aim is the users to realize that they are those feeding the machinery by flooding the web with improved-filtered pictures of themselves or by accepting any clause from various online services. To achieve this, the interaction is structured in three parts that match the steps of creating an online identity: (i) filling personal data, (ii) agreeing to terms and conditions, and (iii) receiving personalised content. The fake body is generated in twelve minutes, along with the complex scene construction. The project offers a new perspective on how to approach technology ethics by involving people in the exploration of artificial intelligence. The decomposition of the frame along with the image simplification, real time rendering and the ethical dimension of the project, are the components that extend the current state of the art. This paper has discussed some of the risks of artificial intelligence along with the proposal of art as a way to approach technology ethics. State of the art technologies are used to produce art that captivates and impresses the general public. The artworks highlight ethical problems that our society is facing, creating a thoughtful space of analysis, experience and debate around the moral concerns in the technological era.

The current work can be extended by artists in the audio dimension and enhance, perfect and reinterpret the scene components in the visual dimension. Also, on the technical side, the distributed architecture can be improved for a faster data transfer, while the GAN can be adapted to support higher resolution images. People could feature themselves in movies, visualise dreams or imagination about themselves and personalise virtually any picture or video. This variety of self-expressive goals could be seen as a gain in personal or artistic autonomy. Since the model of mirror is central in the current art installation, one can introduce the Lacan's paradigm of *imaginary order* and the concept of the *mirror stage* as the model of the mirror that is central to The Profile application.

Acknowledgements Authors would like to thank the anonymous reviewers for their valuable comments and suggestions. The project was improved and further developed during the https://www.youtube.com/watch?v=h9g2lTc5lAoMindBugs project. TheProfile @ MindBugs is part of the https://mediafutures.eu/MediaFutures project. This project has received funding from the European Union's framework Horizon 2020 for research and innovation programme under grant agreement No 951962.

Code Availability Code and data are available at https://github.com/cheresioana/ai-art-TheProfile/tree/master The project was improved and further developed during the MindBugs project https://www.youtube.com/watch?v=h9g2lTc5lAo. TheProfile @ MindBugs is part of the MediaFutures project https://mediafutures.eu/.

Declarations

Conflict of Interests The authors declare they have no conflict of interest.

References

1. Aifanti N, Papachristou C, Delopoulos A (2010) The mug facial expression database. In: 11th International workshop on image analysis for multimedia interactive services WIAMIS 10, pp 1–4. IEEE
2. Baudrillard J (1994) Simulacra and simulation. University of Michigan press
3. Alighieri D (1472) Divine comedy, vol 1
4. Agüera y Arcas B (2017) Art in the age of machine intelligence. In: Arts, vol 6, p 18, multidisciplinary digital publishing institute
5. Baumann F, Lorenz-Spreen P, Sokolov IM, Starnini M (2020) Modeling echo chambers and polarization dynamics in social networks. Phys Rev Lett 124(4):048301
6. Van den Berg P, Paxton SJ, Keery H, Wall M, Guo J, Neumark-Sztainer D (2007) Body dissatisfaction and body comparison with media images in males and females. Body Image 4(3):257–268
7. Bregler C, Covell M, Slaney M (1997) Video rewrite: driving visual speech with audio. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques, pp 353–360
8. Cao Z, Hidalgo G, Simon T, Wei SE, Sheikh Y (2019) Openpose: realtime multi-person 2d pose estimation using part affinity fields. IEEE transactions on pattern analysis and machine intelligence 43(1):172–186
9. Caporusso N (2020) Deepfakes for the good: a beneficial application of contentious artificial intelligence technology. In: International conference on applied human factors and ergonomics, pp 235–241. Springer
10. Carter S, Nielsen M (2017) Using artificial intelligence to augment human intelligence. Distill 2(12):e9
11. Chan C, Ginosar S, Zhou T, Efros A (2019) Everybody dance now. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 5933–5942
12. Chesney B, Citron D (2019) Deep fakes: a looming challenge for privacy, democracy, and national security. Calif L Rev 107:1753
13. Chomanski B (2021) The missing ingredient in the case for regulating big tech. Mind Mach, pp 1–19 https://ec.europa.eu/newsroom/dae/redirection/document/75788. Accessed date 19 Feb 2023
14. Commission E (2021) Proposal for regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts. Tech. rep., European Commission. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX2021PC0206

Bibliography

[CG23] Ioana Cheres and Adrian Groza. The profile: unleashing your deepfake self. *Multimedia Tools and Applications*, pages 1–16, 2023.

[Che22] Ioana Cheres, Mar 2022. https://www.youtube.com/watch?v=5IiwAIx4WwA.

[CST+19] Haochen Chen, Syed Fahad Sultan, Yingtao Tian, Muhao Chen, and Steven Skiena. Fast and accurate network embeddings via very sparse random projection. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 399–408, 2019.

[DML11] Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586, 2011.

[HBC+21] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37, 2021.

[Her87] Frank Herbert. *Heretics of Dune*, volume 5. Penguin, 1987.

[HN22] Amy E Hodler and Mark Needham. Graph data science using neo4j. In *Massive Graph Analytics*, pages 433–457. Chapman and Hall/CRC, 2022.

[JPC+21] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

[LHK15] Hao Lu, Mahantesh Halappanavar, and Ananth Kalyanaraman. Parallel heuristics for scalable community detection. *Parallel Computing*, 47:19–37, 2015.

[LHS17] Huiling Lu, Zhiguo Hong, and Minyong Shi. Analysis of film data based on neo4j. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pages 675–677. IEEE, 2017.

[Min] Mindbugs. Mindbugs. https://mindbugs.ro/.

[Ope23] OpenAI platform documentation, August 2023. https://platform.openai.com/docs/.

[PAG06] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. In *International semantic web conference*, pages 30–43. Springer, 2006.

[Ric07] Leonard Richardson. Beautiful soup documentation, 2007.

[Sin12] Amit Singhal. Introducing the knowledge graph: Things, not strings, May 2012.

[Sub18] Kalpathy Ramaiyer Subramanian. Myth and mystery of shrinking attention span. *International Journal of Trend in Research and Development*, 5(1):1–6, 2018.

[SWL+18] Qi Song, Yinghui Wu, Peng Lin, Luna Xin Dong, and Hui Sun. Mining summaries for knowledge graph search. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1887–1900, 2018.

[TBB21] Ludovic Terren and Rosa Borge-Bravo. Echo chambers on social media: A systematic review of the literature. *Review of Communication Research*, 9:99–118, 2021.

[Tho23] Nicholas Thompson. A conversation with yuval noah harari about artificial intelligence, July 2023. https://www.linkedin.com/pulse/conversation-yuval-noah-harari-artificial-nicholas-thompson/.

[Thr23] Threejs force-directed graph, August 2023. https://github.com/vasturiano/three-forcegraph.

[Tuc22] David Tuck. A cancer graph: a lung cancer property graph database in neo4j. *BMC Research Notes*, 15(1):45, 2022.

[VK14] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[Wik23] Wikidata. Wikidata:sparql tutorial, May 2023. https://www.wikidata.org/wiki/Wikidata:SPARQL$_{t}utorial.$