**IDENTIFYING INAPPROPRIATE LANGUAGE AND HATE SPEECH USING AI**

**A Project Report Submitted in partial fulfilment of the requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

**RUDRARAJU SRIYA 20221A55A5**

**TULA RENU NEELIMA 20221A05B7**

**SALADI SAI SURYA GANESH 20221A05A7**

**UNDURTHI DEERAJ NAGA THARUN 21225A0511**

**Under the Guidance of**

**M. V. Krishna Subash M. Tech**

**Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE**

**(AUTONOMOUS)**

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)

(Accredited by N.B.A & NAAC with ‘A’ Grade)

**ODALAREVU – 533210**

**2020-24**

**BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE**

**(AUTONOMOUS)**

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)

(Accredited by N.B.A & NAAC with ‘A’ Grade)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project work entitled “**IDENTIFYING INAPPROPRIATE LANGUAGE AND HATE SPEECH USING AI**” is being submitted for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering, at BVC Engineering College, Odalarevu**, is a bonafide work done by **RUDRARAJU SRIYA (20221A55A5) , TULA RENU NEELIMA (20221A05B7) , SALADI SAI SURYA GANESH (20221A05A7), UNDURTHI DEERAJ NAGA THARUN (21225A0511)** for the academic year 2023-24 and it has been found suitable for acceptance according to the requirement of University. The results embodied in this thesis have not been submitted to any other University Institute for the award of any degree.

**Project Guide Head of the Department**

**M. V. KRISHNA SUBASH Dr. B S N Murthy**

**M.Tech M.Tech, Ph.D, MISTE**

**Assistant Professor**

**External Examiner**

**ACKNOWLEDGEMENT**

We express our deep sense of gratitude and respect towards our guide **M. V. KRISHNA SUBASH ASSOCIATE PROFESSOR**, BVC Engineering College, Odalarevu for his excellent guidance right from selection of the Project work. His constant encouragement and support have been the cause of our success in completing this thesis in the college.

We are thankful To **Dr. B S N Murthy, Professor and Head of the Department of CSE** for facilitating, training, counselling, and support to carry out the project work.

We are also thankful to **Dr. Chandramouli V S A**, Principal, BVC Engineering College, Odalarevu, for his support during and till the completion of the project.

We are thankful to all the **teaching and non-teaching staff** of Computer Science & Engineering Department, **Management**, BVC Engineering College, Odalarevu, and our friends for their direct and indirect help provided to us in completing the project.

**PROJECT ASSOCIATES**

**RUDRARAJU SRIYA 20221A55A5**

**TULA RENU NEELIMA 20221A05B7**

**SALADI SAI SURYA GANESH 20221A05A7**

**UNDURTHI DEERAJ NAGA THARUN 21225A0511**

**ABSTRACT**

The pervasive nature of online communication has made it a breeding ground for inappropriate language and hate speech, presenting a significant societal challenge. In response, researchers and developers have turned to artificial intelligence (AI) to create automated systems capable of identifying and mitigating such harmful content. This paper provides a comprehensive review of the use of AI in identifying inappropriate language and hate speech, focusing on its application in social media platforms and online forums.

In recent years, the rise of social media platforms has led to an increase in the prevalence of inappropriate language and hate speech online. This has raised concerns about the negative impact such content can have on individuals and communities. In response, researchers and developers have turned to artificial intelligence (AI) techniques to automatically detect and mitigate this harmful content. This paper explores the use of AI for identifying inappropriate language and hate speech, focusing on techniques such as natural language processing (NLP), machine learning, and deep learning. We discuss the challenges associated with detecting inappropriate language and hate speech, including the nuances of language, cultural differences, and evolving forms of hate speech. We also examine the ethical considerations involved in developing and deploying AI systems for this purpose, such as ensuring fairness, transparency, and accountability. Finally, we present case studies and examples of AI systems that have been developed to detect inappropriate language and hate speech, highlighting their effectiveness and limitations. Overall, this paper demonstrates the potential of AI to play a crucial role in addressing the issue of inappropriate language and hate speech online, while also emphasizing the need for responsible development and deployment of AI systems in this domain. This abstract provides a brief overview of the topic, the challenges involved, ethical considerations, and the potential impact of AI in addressing the issue of inappropriate language and hate speech. You can expand on each of these points in your paper to fill out the content for the eight pages.

The paper begins by defining inappropriate language and hate speech, highlighting the complexities involved in their detection, such as sarcasm, slang, and cultural context. It then explores the various AI techniques used for detection, including rule-based systems, machine learning (ML), and deep learning (DL). Special attention is given to the challenges faced in developing AI models for this task, such as the need for large annotated datasets and the risk of algorithmic bias.

Ethical considerations in the development and deployment of AI systems for identifying inappropriate language and hate speech are also discussed, including issues of privacy, free speech, and algorithmic accountability. The paper further examines the impact of AI in addressing online toxicity, citing case studies and real-world examples to demonstrate the effectiveness of AI-driven solutions.

The proliferation of online communication has led to a surge in inappropriate language and hate speech, posing significant challenges for platforms and society at large. In response, researchers and practitioners have turned to artificial intelligence (AI) to develop automated systems for detecting and mitigating such content. This paper provides an in-depth exploration of the use of AI in identifying inappropriate language and hate speech. It begins by discussing the definition and characteristics of inappropriate language and hate speech, emphasizing the importance of context and cultural nuances in their interpretation. The paper then surveys the state-of-the-art AI techniques used for detection, including natural language processing (NLP), machine learning (ML), and deep learning (DL) models. Special attention is given to the challenges inherent in detecting and classifying such content, such as the dynamic nature of language and the subtlety of hate speech. Ethical considerations in the development and deployment of AI systems for this purpose are also examined, including issues of bias, fairness, and privacy. The paper further explores the impact of AI in addressing the proliferation of inappropriate language and hate speech online, highlighting successful case studies and real-world applications. Finally, future directions and open research questions in this field are discussed, emphasizing the need for continued innovation and collaboration to combat online toxicity effectively. Through this comprehensive review, the paper aims to provide insights into the current landscape of AI-driven solutions for identifying inappropriate language and hate speech and to inspire further research and development in this critical area. This abstract sets the stage for an in-depth examination of the topic, touching on key aspects such as definitions, AI techniques, challenges, ethics, impact, and future directions. You can expand on each of these points in your paper to meet the eight-page requirement.

Finally, the paper outlines future research directions in this field, emphasizing the need for improved models that can handle linguistic nuances and evolving forms of hate speech. It concludes by advocating for a multi-stakeholder approach involving researchers, platform developers, and policymakers to develop effective strategies for combating inappropriate language and hate speech online.

# INTRODUCTION

The advent of the internet and social media has revolutionized the way we communicate, allowing people to connect and share information globally. However, this unprecedented level of connectivity has also given rise to new challenges, particularly in the form of inappropriate language and hate speech. Inappropriate language refers to expressions that are offensive, vulgar, or disrespectful, while hate speech encompasses language that incites violence or discrimination against individuals or groups based on attributes such as race, religion, ethnicity, or sexual orientation.

The proliferation of inappropriate language and hate speech online has raised concerns about their impact on individuals, communities, and society at large. Studies have shown that exposure to such content can lead to negative psychological effects, including increased levels of stress, anxiety, and depression. Moreover, hate speech has been linked to real-world violence and discrimination, making its detection and mitigation a matter of urgency.

In response to these challenges, researchers and developers have turned to artificial intelligence (AI) to develop automated systems for identifying and combating inappropriate language and hate speech online. AI offers several advantages for this task, including its ability to process large volumes of text data quickly and its potential for continuous learning and improvement. Additionally, AI systems can operate at scale, allowing them to monitor and analyse content across multiple platforms simultaneously.

This paper provides a comprehensive overview of the use of AI in identifying inappropriate language and hate speech. It begins by discussing the definition and characteristics of inappropriate language and hate speech, highlighting the nuances involved in their detection. The paper then explores the various AI techniques used for this purpose, including natural language processing (NLP), machine learning (ML), and deep learning (DL) models. Special attention is given to the challenges inherent in detecting and classifying such content, such as the dynamic nature of language and the subtlety of hate speech.

Ethical considerations in the development and deployment of AI systems for identifying inappropriate language and hate speech are also examined. These include issues of bias, fairness, and privacy, as well as the broader implications for freedom of speech and censorship. The paper further explores the impact of AI in addressing online toxicity, citing case studies and real-world examples to demonstrate the effectiveness of AI-driven solutions.

Finally, the paper outlines future research directions in this field, highlighting the need for improved AI models that can handle linguistic nuances and evolving forms of hate speech. It also emphasizes the importance of a multi-stakeholder approach involving researchers, platform developers, and policymakers to develop effective strategies for combating inappropriate language and hate speech online.

Through this comprehensive analysis, the paper aims to provide insights into the current state of AI in identifying inappropriate language and hate speech and to serve as a foundation for future research and development in this critical area.

## 1.1 What is Artificial Intelligence (AI)

Artificial intelligence (AI), in its broadest sense, is [intelligence](https://en.wikipedia.org/wiki/Intelligence) exhibited by [machines](https://en.wikipedia.org/wiki/Machine), particularly [computer systems](https://en.wikipedia.org/wiki/Computer_systems). It is a [field of research](https://en.wikipedia.org/wiki/Field_of_research) in [computer science](https://en.wikipedia.org/wiki/Computer_science) that develops and studies methods and software which enable machines to [perceive their environment](https://en.wikipedia.org/wiki/Machine_perception) and uses [learning](https://en.wikipedia.org/wiki/Machine_learning) and intelligence to take actions that maximize their chances of achieving defined goals. Such machines may be called AIs.

[Alan Turing](https://en.wikipedia.org/wiki/Alan_Turing) was the first person to conduct substantial research in the field that he called machine intelligence. Artificial intelligence was founded as an academic discipline in 1956. The field went through multiple cycles of optimism, followed by periods of disappointment and loss of funding, known as [AI winter](https://en.wikipedia.org/wiki/AI_winter). Funding and interest vastly increased after 2012 when [deep learning](https://en.wikipedia.org/wiki/Deep_learning) surpassed all previous AI techniques, and after 2017 with the [transformer architecture](https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)). This led to the [AI boom](https://en.wikipedia.org/wiki/AI_boom) of the early 2020s, with companies, universities, and laboratories overwhelmingly based in the United States pioneering significant [advances in artificial intelligence](https://en.wikipedia.org/wiki/Advances_in_artificial_intelligence).

The growing use of artificial intelligence in the 21st century is influencing [a societal and economic shift](https://en.wikipedia.org/wiki/AI_era) towards increased automation, [data-driven decision-making](https://en.wikipedia.org/wiki/Data-driven_decision-making), and the [integration of AI systems](https://en.wikipedia.org/wiki/Artificial_intelligence_systems_integration) into various economic sectors and areas of life, [impacting job markets](https://en.wikipedia.org/wiki/Workplace_impact_of_artificial_intelligence), [healthcare](https://en.wikipedia.org/wiki/Artificial_intelligence_in_healthcare), government, industry, and [education](https://en.wikipedia.org/wiki/Artificial_intelligence_in_education). This raises questions about [the long-term effects](https://en.wikipedia.org/wiki/AI_aftermath_scenarios), [ethical implications](https://en.wikipedia.org/wiki/Ethics_of_artificial_intelligence), and [risks of AI](https://en.wikipedia.org/wiki/AI_risk), prompting discussions about [regulatory policies](https://en.wikipedia.org/wiki/Regulation_of_artificial_intelligence) to ensure the [safety and benefits of the technology](https://en.wikipedia.org/wiki/AI_safety).

The various sub-fields of AI research are centred around particular goals and the use of particular tools. The traditional goals of AI research include [reasoning](https://en.wikipedia.org/wiki/Automated_reasoning), [knowledge representation](https://en.wikipedia.org/wiki/Knowledge_representation), [planning](https://en.wikipedia.org/wiki/Automated_planning_and_scheduling), [learning](https://en.wikipedia.org/wiki/Machine_learning), [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), perception, and support for [robotics](https://en.wikipedia.org/wiki/Robotics). [General intelligence](https://en.wikipedia.org/wiki/Artificial_general_intelligence) the ability to complete any task performable by a human on an at least equal level is among the field's long-term goals.

To reach these goals, AI researchers have adapted and integrated a wide range of techniques, including [search](https://en.wikipedia.org/wiki/State_space_search) and [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization), [formal logic](https://en.wikipedia.org/wiki/Logic#Formal_logic), [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network), and methods based on [statistics](https://en.wikipedia.org/wiki/Statistics), [operations research](https://en.wikipedia.org/wiki/Operations_research), and [economics](https://en.wikipedia.org/wiki/Economics).AI also draws upon [psychology](https://en.wikipedia.org/wiki/Psychology), [linguistics](https://en.wikipedia.org/wiki/Linguistics), [philosophy](https://en.wikipedia.org/wiki/Philosophy_of_artificial_intelligence), [neuroscience](https://en.wikipedia.org/wiki/Neuroscience), and other fields.

## 1.2 Goals

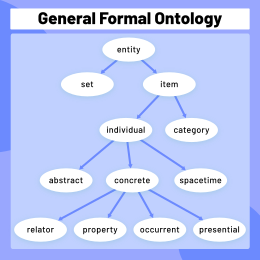
The general problem of simulating (or creating) intelligence has been broken into sub-problems. These consist of traits or capabilities that researchers expect an intelligent system to display. The traits described below have received the most attention and cover the scope of AI research.

## 1.21 Reasoning and problem solving

Early researchers developed algorithms that imitated step-by-step reasoning that humans use when they solve puzzles or make logical [deductions](https://en.wikipedia.org/wiki/Deductive_reasoning). By the late 1980s and 1990s, methods were developed for dealing with [uncertain](https://en.wikipedia.org/wiki/Uncertainty) or incomplete information, employing concepts from [probability](https://en.wikipedia.org/wiki/Probability) and [economics](https://en.wikipedia.org/wiki/Economics).

Many of these algorithms are insufficient for solving large reasoning problems because they experience a "combinatorial explosion": they became exponentially slower as the problems grew larger. Even humans rarely use the step-by-step deduction that early AI research could model. They solve most of their problems using fast, intuitive judgments. Accurate and efficient reasoning is an unsolved problem.

## 1.22 Knowledge representation

[](https://en.wikipedia.org/wiki/File:General_Formal_Ontology.svg)

An ontology represents knowledge as a set of concepts within a domain and the relationships between those concepts. [Knowledge representation](https://en.wikipedia.org/wiki/Knowledge_representation) and [knowledge engineering](https://en.wikipedia.org/wiki/Knowledge_engineering) allow AI programs to answer questions intelligently and make deductions about real-world facts. Formal knowledge representations are used in content-based indexing and retrieval, scene interpretation, clinical decision support, knowledge discovery (mining "interesting" and actionable inferences from large [databases](https://en.wikipedia.org/wiki/Database)), and other areas.

A [knowledge base](https://en.wikipedia.org/wiki/Knowledge_base) is a body of knowledge represented in a form that can be used by a program. An [ontology](https://en.wikipedia.org/wiki/Ontology_(information_science)) is the set of objects, relations, concepts, and properties used by a particular domain of knowledge. Knowledge bases need to represent things such as: objects, properties, categories and relations between objects; situations, events, states and time; causes and effects; knowledge about knowledge (what we know about what other people know); [default reasoning](https://en.wikipedia.org/wiki/Default_reasoning) (things that humans assume are true until they are told differently and will remain true even when other facts are changing); and many other aspects and domains of knowledge.

Among the most difficult problems in knowledge representation are: the breadth of commonsense knowledge (the set of atomic facts that the average person knows is enormous);  and the sub-symbolic form of most commonsense knowledge (much of what people know is not represented as "facts" or "statements" that they could express verbally). There is also the difficulty of [knowledge acquisition](https://en.wikipedia.org/wiki/Knowledge_acquisition), the problem of obtaining knowledge for AI applications.

### 1.23 Planning and decision making

An "agent" is anything that perceives and takes actions in the world. A [rational agent](https://en.wikipedia.org/wiki/Rational_agent) has goals or preferences and takes actions to make them happen. In [automated planning](https://en.wikipedia.org/wiki/Automated_planning_and_scheduling), the agent has a specific goal. In [automated decision making](https://en.wikipedia.org/wiki/Automated_decision_making), the agent has preferences—there are some situations it would prefer to be in, and some situations it is trying to avoid. The decision making agent assigns a number to each situation (called the "[utility](https://en.wikipedia.org/wiki/Utility_(economics))") that measures how much the agent prefers it. For each possible action, it can calculate the "[expected utility](https://en.wikipedia.org/wiki/Expected_utility)": the [utility](https://en.wikipedia.org/wiki/Utility) of all possible outcomes of the action, weighted by the probability that the outcome will occur. It can then choose the action with the maximum expected utility.

In [classical planning](https://en.wikipedia.org/wiki/Automated_planning_and_scheduling#classical_planning), the agent knows exactly what the effect of any action will be. In most real-world problems, however, the agent may not be certain about the situation they are in (it is "unknown" or "unobservable") and it may not know for certain what will happen after each possible action (it is not "deterministic"). It must choose an action by making a probabilistic guess and then reassess the situation to see if the action worked.

In some problems, the agent's preferences may be uncertain, especially if there are other agents or humans involved. These can be learned (e.g., with [inverse reinforcement learning](https://en.wikipedia.org/wiki/Inverse_reinforcement_learning)) or the agent can seek information to improve its preferences. [Information value theory](https://en.wikipedia.org/wiki/Information_value_theory) can be used to weigh the value of exploratory or experimental actions. The space of possible future actions and situations is typically [intractably](https://en.wikipedia.org/wiki/Intractable_problem) large, so the agents must take actions and evaluate situations while being uncertain what the outcome will be.

A [Markov decision process](https://en.wikipedia.org/wiki/Markov_decision_process) has a [transition model](https://en.wikipedia.org/wiki/Finite-state_machine) that describes the probability that a particular action will change the state in a particular way, and a [reward function](https://en.wikipedia.org/wiki/Reward_function) that supplies the utility of each state and the cost of each action. A [policy](https://en.wikipedia.org/wiki/Reinforcement_learning#Policy) associates a decision with each possible state. The policy could be calculated (e.g., by [iteration](https://en.wikipedia.org/wiki/Policy_iteration)), be [heuristic](https://en.wikipedia.org/wiki/Heuristic), or it can be learned.

[Game theory](https://en.wikipedia.org/wiki/Game_theory) describes rational behavior of multiple interacting agents, and is used in AI programs that make decisions that involve other agents.

### 1.24 Learning

[Machine learning](https://en.wikipedia.org/wiki/Machine_learning) is the study of programs that can improve their performance on a given task automatically. It has been a part of AI from the beginning.

There are several kinds of machine learning. [Unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning) analyses a stream of data and finds patterns and makes predictions without any other guidance. [Supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) requires a human to label the input data first, and comes in two main varieties: [classification](https://en.wikipedia.org/wiki/Statistical_classification) (where the program must learn to predict what category the input belongs in) and [regression](https://en.wikipedia.org/wiki/Regression_analysis) (where the program must deduce a numeric function based on numeric input).

In [reinforcement learning](https://en.wikipedia.org/wiki/Reinforcement_learning) the agent is rewarded for good responses and punished for bad ones. The agent learns to choose responses that are classified as "good".[]](https://en.wikipedia.org/wiki/Artificial_intelligence#cite_note-54) [Transfer learning](https://en.wikipedia.org/wiki/Transfer_learning) is when the knowledge gained from one problem is applied to a new problem. [Deep learning](https://en.wikipedia.org/wiki/Deep_learning) is a type of machine learning that runs inputs through biologically inspired [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_networks) for all of these types of learning.

[Computational learning theory](https://en.wikipedia.org/wiki/Computational_learning_theory) can assess learners by [computational complexity](https://en.wikipedia.org/wiki/Computational_complexity), by [sample complexity](https://en.wikipedia.org/wiki/Sample_complexity) (how much data is required), or by other notions of [optimization](https://en.wikipedia.org/wiki/Optimization_theory).

### 1.25 Natural language processing

[Natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing) (NLP) allows programs to read, write and communicate in human languages such as [English](https://en.wikipedia.org/wiki/English_(language)). Specific problems include [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), [speech synthesis](https://en.wikipedia.org/wiki/Speech_synthesis), [machine translation](https://en.wikipedia.org/wiki/Machine_translation), [information extraction](https://en.wikipedia.org/wiki/Information_extraction), [information retrieval](https://en.wikipedia.org/wiki/Information_retrieval) and [question answering](https://en.wikipedia.org/wiki/Question_answering).

Early work, based on [Noam Chomsky](https://en.wikipedia.org/wiki/Noam_Chomsky)'s [generative grammar](https://en.wikipedia.org/wiki/Generative_grammar) and [semantic networks](https://en.wikipedia.org/wiki/Semantic_network), had difficulty with [word-sense disambiguation](https://en.wikipedia.org/wiki/Word-sense_disambiguation) unless restricted to small domains called "[micro-worlds](https://en.wikipedia.org/wiki/Blocks_world)" (due to the common sense knowledge problem). [Margaret Masterman](https://en.wikipedia.org/wiki/Margaret_Masterman) believed that it was meaning and not grammar that was the key to understanding languages, and that [thesauri](https://en.wikipedia.org/wiki/Thesauri) and not dictionaries should be the basis of computational language structure.

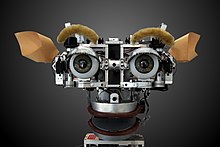
Modern deep learning techniques for NLP include [word embedding](https://en.wikipedia.org/wiki/Word_embedding) (representing words, typically as [vectors](https://en.wikipedia.org/wiki/Vector_space) encoding their meaning), [transformers](https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)) (a deep learning architecture using an [attention](https://en.wikipedia.org/wiki/Attention_(machine_learning)) mechanism), and others. In 2019, [generative pre-trained transformer](https://en.wikipedia.org/wiki/Generative_pre-trained_transformer) (or "GPT") language models began to generate coherent text, and by 2023 these models were able to get human-level scores on the [bar exam](https://en.wikipedia.org/wiki/Bar_exam), [SAT](https://en.wikipedia.org/wiki/Scholastic_aptitude_test) test, [GRE](https://en.wikipedia.org/wiki/Graduate_Record_Examinations) test, and many other real-world applications.

### 1.26 Perception

[Machine perception](https://en.wikipedia.org/wiki/Machine_perception) is the ability to use input from sensors (such as cameras, microphones, wireless signals, active [lidar](https://en.wikipedia.org/wiki/Lidar), sonar, radar, and [tactile sensors](https://en.wikipedia.org/wiki/Tactile_sensor)) to deduce aspects of the world. [Computer vision](https://en.wikipedia.org/wiki/Computer_vision) is the ability to analyze visual input.

The field includes [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), [image classification](https://en.wikipedia.org/wiki/Image_classification), [facial recognition](https://en.wikipedia.org/wiki/Facial_recognition_system), [object recognition](https://en.wikipedia.org/wiki/Object_recognition), and [robotic perception](https://en.wikipedia.org/wiki/Robotic_sensing).

### Social intelligence

[](https://en.wikipedia.org/wiki/File:Kismet-IMG_6007-gradient.jpg)

[Kismet](https://en.wikipedia.org/wiki/Kismet_(robot)), a robot head which was made in the 1990s; a machine that can recognize and simulate emotions. [Affective computing](https://en.wikipedia.org/wiki/Affective_computing) is an interdisciplinary umbrella that comprises systems that recognize, interpret, process or simulate human [feeling, emotion and mood](https://en.wikipedia.org/wiki/Affect_(psychology)). For example, some [virtual assistants](https://en.wikipedia.org/wiki/Virtual_assistant) are programmed to speak conversationally or even to banter humorously; it makes them appear more sensitive to the emotional dynamics of human interaction, or to otherwise facilitate [human–computer interaction](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction).

However, this tends to give naïve users an unrealistic conception of the intelligence of existing computer agents. Moderate successes related to affective computing include textual [sentiment analysis](https://en.wikipedia.org/wiki/Sentiment_analysis) and, more recently, [multimodal sentiment analysis](https://en.wikipedia.org/wiki/Multimodal_sentiment_analysis), wherein AI classifies the affects displayed by a videotaped subject.

### 1.27 General intelligence

A machine with [artificial general intelligence](https://en.wikipedia.org/wiki/Artificial_general_intelligence) should be able to solve a wide variety of problems with breadth and versatility similar to human intelligence.

**1.3 Techniques**

AI research uses a wide variety of techniques to accomplish the goals above.

### 1.31 Search and optimization

AI can solve many problems by intelligently searching through many possible solutions. There are two very different kinds of search used in AI: [state space search](https://en.wikipedia.org/wiki/State_space_search) and [local search](https://en.wikipedia.org/wiki/Local_search_(optimization)).

#### 1.32 State space search

[State space search](https://en.wikipedia.org/wiki/State_space_search) searches through a tree of possible states to try to find a goal state. For example, [planning](https://en.wikipedia.org/wiki/Automated_planning_and_scheduling) algorithms search through trees of goals and subgoals, attempting to find a path to a target goal, a process called [means-ends analysis](https://en.wikipedia.org/wiki/Means-ends_analysis).

[Simple exhaustive searches](https://en.wikipedia.org/wiki/Brute_force_search) are rarely sufficient for most real-world problems: the [search space](https://en.wikipedia.org/wiki/Search_algorithm) (the number of places to search) quickly grows to [astronomical numbers](https://en.wikipedia.org/wiki/Astronomically_large). The result is a search that is [too slow](https://en.wikipedia.org/wiki/Computation_time) or never completes. "[Heuristics](https://en.wikipedia.org/wiki/Heuristics)" or "rules of thumb" can help to prioritize choices that are more likely to reach a goal.

[Adversarial search](https://en.wikipedia.org/wiki/Adversarial_search) is used for [game-playing](https://en.wikipedia.org/wiki/Game_AI) programs, such as chess or Go. It searches through a [tree](https://en.wikipedia.org/wiki/Game_tree) of possible moves and counter-moves, looking for a winning position.

#### 1.33 Local search

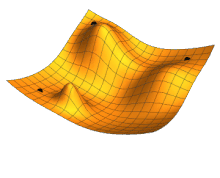
[](https://en.wikipedia.org/wiki/File:Gradient_descent.gif)

Illustration of [gradient descent](https://en.wikipedia.org/wiki/Gradient_descent) for 3 different starting points. Two parameters (represented by the plan coordinates) are adjusted in order to minimize the [loss function](https://en.wikipedia.org/wiki/Loss_function) (the height). [Local search](https://en.wikipedia.org/wiki/Local_search_(optimization)) uses [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization) to find a solution to a problem. It begins with some form of guess and refines it incrementally.

[Gradient descent](https://en.wikipedia.org/wiki/Gradient_descent) is a type of local search that optimizes a set of numerical parameters by incrementally adjusting them to minimize a [loss function](https://en.wikipedia.org/wiki/Loss_function). Variants of gradient descent are commonly used to train neural networks.

Another type of local search is [evolutionary computation](https://en.wikipedia.org/wiki/Evolutionary_computation), which aims to iteratively improve a set of candidate solutions by "mutating" and "recombining" them, [selecting](https://en.wikipedia.org/wiki/Artificial_selection) only the fittest to survive each generation.

Distributed search processes can coordinate via [swarm intelligence](https://en.wikipedia.org/wiki/Swarm_intelligence) algorithms. Two popular swarm algorithms used in search are [particle swarm optimization](https://en.wikipedia.org/wiki/Particle_swarm_optimization) (inspired by bird [flocking](https://en.wikipedia.org/wiki/Flocking_(behavior))) and [ant colony optimization](https://en.wikipedia.org/wiki/Ant_colony_optimization) (inspired by [ant trails](https://en.wikipedia.org/wiki/Ant_trail)).

### 1.34 Logic

Formal [logic](https://en.wikipedia.org/wiki/Logic) is used for [reasoning](https://en.wikipedia.org/wiki/Automatic_reasoning) and [knowledge representation](https://en.wikipedia.org/wiki/Knowledge_representation). Formal logic comes in two main forms: [propositional logic](https://en.wikipedia.org/wiki/Propositional_logic) (which operates on statements that are true or false and uses [logical connectives](https://en.wikipedia.org/wiki/Logical_connective) such as "and", "or", "not" and "implies") and [predicate logic](https://en.wikipedia.org/wiki/Predicate_logic) (which also operates on objects, predicates and relations and uses [quantifiers](https://en.wikipedia.org/wiki/Quantifier_(logic)) such as "Every X is a Y" and "There are some Xs that are Ys").

[Deductive reasoning](https://en.wikipedia.org/wiki/Deductive_reasoning) in logic is the process of [proving](https://en.wikipedia.org/wiki/Logical_proof) a new statement ([conclusion](https://en.wikipedia.org/wiki/Logical_consequence)) from other statements that are given and assumed to be true (the [premises](https://en.wikipedia.org/wiki/Premise)). Proofs can be structured as proof [trees](https://en.wikipedia.org/wiki/Tree_structure), in which nodes are labelled by sentences, and children nodes are connected to parent nodes by [inference rules](https://en.wikipedia.org/wiki/Inference_rule).

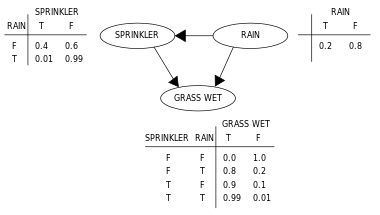
Given a problem and a set of premises, problem-solving reduces to searching for a proof tree whose root node is labelled by a solution of the problem and whose leaf nodes are labelled by premises or [axioms](https://en.wikipedia.org/wiki/Axiom). In the case of [Horn clauses](https://en.wikipedia.org/wiki/Horn_clause), problem-solving search can be performed by reasoning [forwards](https://en.wikipedia.org/wiki/Forward_chaining) from the premises or [backwards](https://en.wikipedia.org/wiki/Backward_chaining) from the problem. In the more general case of the clausal form of [first-order logic](https://en.wikipedia.org/wiki/First-order_logic), [resolution](https://en.wikipedia.org/wiki/Resolution_(logic)) is a single, axiom-free rule of inference, in which a problem is solved by proving a contradiction from premises that include the negation of the problem to be solved.

Inference in both Horn clause logic and first-order logic is [undecidable](https://en.wikipedia.org/wiki/Undecidable_problem), and therefore [intractable](https://en.wikipedia.org/wiki/Intractable_problem). However, backward reasoning with Horn clauses, which underpins computation in the [logic programming](https://en.wikipedia.org/wiki/Logic_programming) language  prolong, is [Turing complete](https://en.wikipedia.org/wiki/Turing_completeness). Moreover, its efficiency is competitive with computation in other [symbolic programming](https://en.wikipedia.org/wiki/Symbolic_programming) languages.

[Fuzzy logic](https://en.wikipedia.org/wiki/Fuzzy_logic) assigns a "degree of truth" between 0 and 1. It can therefore handle propositions that are vague and partially true.

[Non-monotonic logics](https://en.wikipedia.org/wiki/Non-monotonic_logic), including logic programming with [negation as failure](https://en.wikipedia.org/wiki/Negation_as_failure), are designed to handle [default reasoning](https://en.wikipedia.org/wiki/Default_reasoning). Other specialized versions of logic have been developed to describe many complex domains.

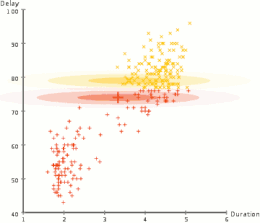
### 1.35 Probabilistic methods for uncertain reasoning

[](https://en.wikipedia.org/wiki/File:SimpleBayesNet.svg)

A simple [Bayesian network](https://en.wikipedia.org/wiki/Bayesian_network), with the associated [conditional probability tables](https://en.wikipedia.org/wiki/Conditional_probability_table)

Many problems in AI (including in reasoning, planning, learning, perception, and robotics) require the agent to operate with incomplete or uncertain information. AI researchers have devised a number of tools to solve these problems using methods from [probability](https://en.wikipedia.org/wiki/Probability) theory and economics. Precise mathematical tools have been developed that analyze how an agent can make choices and plan, using [decision theory](https://en.wikipedia.org/wiki/Decision_theory), [decision analysis](https://en.wikipedia.org/wiki/Decision_analysis), and [information value theory](https://en.wikipedia.org/wiki/Information_value_theory). These tools include models

Probabilistic algorithms can also be used for filtering, prediction, smoothing and finding explanations for streams of data, helping [perception](https://en.wikipedia.org/wiki/Machine_perception) systems to analyze processes that occur over time (e.g., [hidden Markov models](https://en.wikipedia.org/wiki/Hidden_Markov_model) or [Kalman filters](https://en.wikipedia.org/wiki/Kalman_filter)).

[](https://en.wikipedia.org/wiki/File:EM_Clustering_of_Old_Faithful_data.gif)

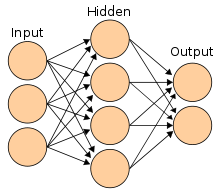
[Expectation-maximization](https://en.wikipedia.org/wiki/Expectation-maximization) [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) of [Old Faithful](https://en.wikipedia.org/wiki/Old_Faithful) eruption data starts from a random guess but then successfully converges on an accurate clustering of the two physically distinct modes of eruption.

### 1.36 Classifiers and statistical learning methods

The simplest AI applications can be divided into two types: classifiers (e.g., "if shiny then diamond"), on one hand, and controllers (e.g., "if diamond then pick up"), on the other hand. [Classifiers](https://en.wikipedia.org/wiki/Classifier_(mathematics)) are functions that use [pattern matching](https://en.wikipedia.org/wiki/Pattern_matching) to determine the closest match. They can be fine-tuned based on chosen examples using [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning). Each pattern (also called an "[observation](https://en.wikipedia.org/wiki/Random_variate)") is labeled with a certain predefined class. All the observations combined with their class labels are known as a [data set](https://en.wikipedia.org/wiki/Data_set). When a new observation is received, that observation is classified based on previous experience.

There are many kinds of classifiers in use. The [decision tree](https://en.wikipedia.org/wiki/Decision_tree) is the simplest and most widely used symbolic machine learning algorithm.  [K-nearest neighbor](https://en.wikipedia.org/wiki/K-nearest_neighbor) algorithm was the most widely used analogical AI until the mid-1990s, and [Kernel methods](https://en.wikipedia.org/wiki/Kernel_methods) such as the [support vector machine](https://en.wikipedia.org/wiki/Support_vector_machine) (SVM) displaced k-nearest neighbor in the 1990s. The [naive Bayes classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier) is reportedly the "most widely used learner" at Google, due in part to its scalability. [Neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) are also used as classifiers.

### 1.37 Artificial neural networks

[](https://en.wikipedia.org/wiki/File:Artificial_neural_network.svg)

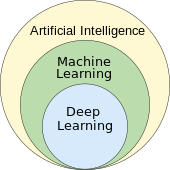
A neural network is an interconnected group of nodes, akin to the vast network of [neurons](https://en.wikipedia.org/wiki/Neuron) in the [human brain](https://en.wikipedia.org/wiki/Human_brain). An artificial neural network is based on a collection of nodes also known as [artificial neurons](https://en.wikipedia.org/wiki/Artificial_neurons), which loosely model the [neurons](https://en.wikipedia.org/wiki/Neurons) in a biological brain. It is trained to recognise patterns; once trained, it can recognise those patterns in fresh data. There is an input, at least one hidden layer of nodes and an output. Each node applies a function and once the [weight](https://en.wikipedia.org/wiki/Weighting) crosses its specified threshold, the data is transmitted to the next layer. A network is typically called a deep neural network if it has at least 2 hidden layers.

Learning algorithms for neural networks use [local search](https://en.wikipedia.org/wiki/Local_search_(optimization)) to choose the weights that will get the right output for each input during training. The most common training technique is the [backpropagation](https://en.wikipedia.org/wiki/Backpropagation) algorithm. Neural networks learn to model complex relationships between inputs and outputs and [find patterns](https://en.wikipedia.org/wiki/Pattern_recognition) in data. In theory, a neural network can learn any function.

In [feedforward neural networks](https://en.wikipedia.org/wiki/Feedforward_neural_network) the signal passes in only one direction. [Recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_network) feed the output signal back into the input, which allows short-term memories of previous input events. [Long short term memory](https://en.wikipedia.org/wiki/Long_short_term_memory) is the most successful network architecture for recurrent networks. [Perceptrons](https://en.wikipedia.org/wiki/Perceptron) use only a single layer of neurons, deep learning uses multiple layers. [Convolutional neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_network) strengthen the connection between neurons that are "close" to each other—this is especially important in [image processing](https://en.wikipedia.org/wiki/Image_processing), where a local set of neurons must [identify an "edge"](https://en.wikipedia.org/wiki/Edge_detection) before the network can identify an object.

### 1.38 Deep learning

### 

[](https://en.wikipedia.org/wiki/File:AI_hierarchy.svg)

Deep learning uses several layers of neurons between the network's inputs and outputs. The multiple layers can progressively extract higher-level features from the raw input. For example, in [image processing](https://en.wikipedia.org/wiki/Image_processing), lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

Deep learning has profoundly improved the performance of programs in many important subfields of artificial intelligence, including [computer vision](https://en.wikipedia.org/wiki/Computer_vision), [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), [image classification](https://en.wikipedia.org/wiki/Image_classification) and others. The reason that deep learning performs so well in so many applications is not known as of 2023. The sudden success of deep learning in 2012–2015 did not occur because of some new discovery or theoretical breakthrough (deep neural networks and backpropagation had been described by many people, as far back as the 1950s) but because of two factors: the incredible increase in computer power (including the hundred-fold increase in speed by switching to [GPUs](https://en.wikipedia.org/wiki/Graphics_processing_units)) and the availability of vast amounts of training data, especially the giant [curated datasets](https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research) used for benchmark testing, such as [ImageNet](https://en.wikipedia.org/wiki/ImageNet).

### 1.39 GPT

[Generative pre-trained transformers](https://en.wikipedia.org/wiki/Generative_pre-trained_transformer) (GPT) are [large language models](https://en.wikipedia.org/wiki/Large_language_model) that are based on the semantic relationships between words in sentences ([natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing)). Text-based GPT models are pre-trained on a large corpus of text which can be from the internet. The pre-training consists in predicting the next [token](https://en.wikipedia.org/wiki/Lexical_analysis) (a token being usually a word, subword, or punctuation). Throughout this pre-training, GPT models accumulate knowledge about the world, and can then generate human-like text by repeatedly predicting the next token. Typically, a subsequent training phase makes the model more truthful, useful and harmless, usually with a technique called [reinforcement learning from human feedback](https://en.wikipedia.org/wiki/Reinforcement_learning_from_human_feedback) (RLHF). Current GPT models are still prone to generating falsehoods called "[hallucinations](https://en.wikipedia.org/wiki/Hallucination_(artificial_intelligence))", although this can be reduced with RLHF and quality data. They are used in [chatbots](https://en.wikipedia.org/wiki/Chatbot), which allow you to ask a question or request a task in simple text.

Current models and services include:

[Gemini](https://en.wikipedia.org/wiki/Gemini_(chatbot)) (formerlyBard), [ChatGPT](https://en.wikipedia.org/wiki/ChatGPT), [Grok](https://en.wikipedia.org/wiki/Grok_(chatbot)), [Claude](https://en.wikipedia.org/wiki/Anthropic#Claude), [Copilot](https://en.wikipedia.org/wiki/Microsoft_Copilot) and [LLaMA](https://en.wikipedia.org/wiki/LLaMA). [Multimodal](https://en.wikipedia.org/wiki/Multimodal_learning) GPT models can process different types of data ([modalities](https://en.wikipedia.org/wiki/Modality_(human%E2%80%93computer_interaction))) such as images, videos, sound and text.

### 1.310 Specialized hardware and software

Main articles: [Programming languages for artificial intelligence](https://en.wikipedia.org/wiki/Programming_languages_for_artificial_intelligence) and [Hardware for artificial intelligence](https://en.wikipedia.org/wiki/Hardware_for_artificial_intelligence). In the late 2010s, [graphics processing units](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPUs) that were increasingly designed with AI-specific enhancements and used with specialized [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow) software, had replaced previously used [central processing unit](https://en.wikipedia.org/wiki/Central_processing_unit) (CPUs) as the dominant means for large-scale (commercial and academic) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) models' training. Historically, specialized languages, such as [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)), [Prolog](https://en.wikipedia.org/wiki/Prolog" \o "Prolog), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and others, had been used.

**1.4 Applications**

AI and machine learning technology is used in most of the essential applications of the 2020s, including: [search engines](https://en.wikipedia.org/wiki/Search_engines) (such as [Google Search](https://en.wikipedia.org/wiki/Google_Search)), [targeting online advertisements](https://en.wikipedia.org/wiki/Targeted_advertising), [recommendation systems](https://en.wikipedia.org/wiki/Recommender_system) (offered by [Netflix](https://en.wikipedia.org/wiki/Netflix), [YouTube](https://en.wikipedia.org/wiki/YouTube) or [Amazon](https://en.wikipedia.org/wiki/Amazon_(company))), driving [internettraffic](https://en.wikipedia.org/wiki/Internet_traffic), [targetedadvertising](https://en.wikipedia.org/wiki/Marketing_and_artificial_intelligence) ([AdSense](https://en.wikipedia.org/wiki/AdSense), [Facebook](https://en.wikipedia.org/wiki/Facebook)),[virtualassistants](https://en.wikipedia.org/wiki/Virtual_assistant), [autonomousvehicles](https://en.wikipedia.org/wiki/Autonomous_vehicles) (including [drones](https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle), [ADAS](https://en.wikipedia.org/wiki/Advanced_driver-assistance_system) and [self-driving cars](https://en.wikipedia.org/wiki/Self-driving_cars)), [automatic language translation](https://en.wikipedia.org/wiki/Machine_translation) ([MicrosoftTranslator](https://en.wikipedia.org/wiki/Microsoft_Translator), [GoogleTranslate](https://en.wikipedia.org/wiki/Google_Translate)), [facialrecognition](https://en.wikipedia.org/wiki/Facial_recognition_system) ([Apple](https://en.wikipedia.org/wiki/Apple_Computer)'s [FaceID](https://en.wikipedia.org/wiki/Face_ID) or [Microsoft](https://en.wikipedia.org/wiki/Microsoft)'s [DeepFace](https://en.wikipedia.org/wiki/DeepFace) and [Google](https://en.wikipedia.org/wiki/Google)'s [FaceNet](https://en.wikipedia.org/wiki/FaceNet" \o "FaceNet))and [image labeling](https://en.wikipedia.org/wiki/Automatic_image_annotation) (used by [Facebook](https://en.wikipedia.org/wiki/Facebook), Apple's [iPhoto](https://en.wikipedia.org/wiki/IPhoto) and [TikTok](https://en.wikipedia.org/wiki/TikTok)).

### 1.41 Health and medicine

The application of AI in [medicine](https://en.wikipedia.org/wiki/Medicine) and [medical research](https://en.wikipedia.org/wiki/Medical_research) has the potential to increase patient care and quality of life. Through the lens of the [Hippocratic Oath](https://en.wikipedia.org/wiki/Hippocratic_Oath), medical professionals are ethically compelled to use AI, if applications can more accurately diagnose and treat patients.

For medical research, AI is an important tool for processing and integrating [big data](https://en.wikipedia.org/wiki/Big_data). This is particularly important for [organoid](https://en.wikipedia.org/wiki/Organoid) and [tissue engineering](https://en.wikipedia.org/wiki/Tissue_engineering) development which use [microscopy](https://en.wikipedia.org/wiki/Microscopy) imaging as a key technique in fabrication.  It has been suggested that AI can overcome discrepancies in funding allocated to different fields of research. New AI tools can deepen the understanding of biomedically relevant pathways. For example, [AlphaFold 2](https://en.wikipedia.org/wiki/AlphaFold_2) (2021) demonstrated the ability to approximate, in hours rather than months, the 3D [structure of a protein](https://en.wikipedia.org/wiki/Protein_structure). In 2023, it was reported that AI-guided drug discovery helped find a class of antibiotics capable of killing two different types of drug-resistant bacteria. In 2024, researchers used machine learning to accelerate the search for [Parkinson's disease](https://en.wikipedia.org/wiki/Parkinson%27s_disease) drug treatments. Their aim was to identify compounds that block the clumping, or aggregation, of [alpha-synuclein](https://en.wikipedia.org/wiki/Alpha-synuclein) (the protein that characterises Parkinson's disease). They were able to speed up the initial screening process ten-fold, and to reduce the cost by a thousand-fold.

### 1.42 Games

[Game playing](https://en.wikipedia.org/wiki/Game_AI) programs have been used since the 1950s to demonstrate and test AI's most advanced techniques. [Deep Blue](https://en.wikipedia.org/wiki/IBM_Deep_Blue) became the first computer chess-playing system to beat a reigning world chess champion, [Garry Kasparov](https://en.wikipedia.org/wiki/Garry_Kasparov), on 11 May 1997. In 2011, in a [Jeopardy!](https://en.wikipedia.org/wiki/Jeopardy!) [quiz show](https://en.wikipedia.org/wiki/Quiz_show) exhibition match, [IBM](https://en.wikipedia.org/wiki/IBM)'s [question answering system](https://en.wikipedia.org/wiki/Question_answering_system), [Watson](https://en.wikipedia.org/wiki/Watson_(artificial_intelligence_software)), defeated the two greatest Jeopardy! champions, [Brad Rutter](https://en.wikipedia.org/wiki/Brad_Rutter) and [Ken Jennings](https://en.wikipedia.org/wiki/Ken_Jennings), by a significant margin.  In March 2016, [AlphaGo](https://en.wikipedia.org/wiki/AlphaGo) won 4 out of 5 games of [Go](https://en.wikipedia.org/wiki/Go_(game)) in a match with Go champion [Lee Sedol](https://en.wikipedia.org/wiki/Lee_Sedol), becoming the first [computer Go](https://en.wikipedia.org/wiki/Computer_Go)-playing system to beat a professional Go player without [handicaps](https://en.wikipedia.org/wiki/Go_handicaps). Then in 2017 it [defeated Ke Jie](https://en.wikipedia.org/wiki/AlphaGo_versus_Ke_Jie), who was the best Go player in the world. Other programs handle [imperfect-information](https://en.wikipedia.org/wiki/Imperfect_information) games, such as the [poker](https://en.wikipedia.org/wiki/Poker)-playing program [Pluribus](https://en.wikipedia.org/wiki/Pluribus_(poker_bot)).  [DeepMind](https://en.wikipedia.org/wiki/DeepMind) developed increasingly generalistic [reinforcement learning](https://en.wikipedia.org/wiki/Reinforcement_learning) models, such as with [MuZero](https://en.wikipedia.org/wiki/MuZero" \o "MuZero), which could be trained to play chess, Go, or [Atari](https://en.wikipedia.org/wiki/Atari) games. In 2019, DeepMind's AlphaStar achieved grandmaster level in [StarCraft II](https://en.wikipedia.org/wiki/StarCraft_II), a particularly challenging real-time strategy game that involves incomplete knowledge of what happens on the map. In 2021, an AI agent competed in a PlayStation Gran Turismo competition, winning against four of the world's best Gran Turismo drivers using deep reinforcement learning.

### 1.44 Military

Various countries are deploying AI military applications. The main applications enhance [command and control](https://en.wikipedia.org/wiki/Command_and_control), communications, sensors, integration and interoperability. Research is targeting intelligence collection and analysis, logistics, cyber operations, information operations, and semiautonomous and [autonomous vehicles](https://en.wikipedia.org/wiki/Vehicular_automation). AI technologies enable coordination of sensors and effectors, threat detection and identification, marking of enemy positions, [target acquisition](https://en.wikipedia.org/wiki/Target_acquisition), coordination and deconfliction of distributed [Joint Fires](https://en.wikipedia.org/wiki/Forward_observers_in_the_U.S._military) between networked combat vehicles involving manned and unmanned teams. AI was incorporated into military operations in Iraq and Syria.

In November 2023, US Vice President [Kamala Harris](https://en.wikipedia.org/wiki/Kamala_Harris) disclosed a declaration signed by 31 nations to set guardrails for the military use of AI. The commitments include using legal reviews to ensure the compliance of military AI with international laws, and being cautious and transparent in the development of this technology.

### 1.44 Generative AI

### 

[](https://en.wikipedia.org/wiki/File:Vincent_van_Gogh_in_watercolour.png)

In the early 2020s, [generative AI](https://en.wikipedia.org/wiki/Generative_AI) gained widespread prominence. In March 2023, 58% of U.S. adults had heard about [ChatGPT](https://en.wikipedia.org/wiki/ChatGPT) and 14% had tried it. The increasing realism and ease-of-use of AI-based [text-to-image](https://en.wikipedia.org/wiki/Text-to-image_model) generators such as [Midjourney](https://en.wikipedia.org/wiki/Midjourney), [DALL-E](https://en.wikipedia.org/wiki/DALL-E), and [Stable Diffusion](https://en.wikipedia.org/wiki/Stable_Diffusion) sparked a trend of [viral](https://en.wikipedia.org/wiki/Viral_phenomenon) AI-generated photos. Widespread attention was gained by a fake photo of [Pope Francis](https://en.wikipedia.org/wiki/Pope_Francis) wearing a white puffer coat, the fictional arrest of [Donald Trump](https://en.wikipedia.org/wiki/Donald_Trump), and a hoax of an attack on the [Pentagon](https://en.wikipedia.org/wiki/The_Pentagon), as well as the usage in professional creative arts.

### 1.45 Industry-specific tasks

There are also thousands of successful AI applications used to solve specific problems for specific industries or institutions. In a 2017 survey, one in five companies reported they had incorporated "AI" in some offerings or processes. A few examples are [energy storage](https://en.wikipedia.org/wiki/Energy_storage), medical diagnosis, military logistics, applications that predict the result of judicial decisions, [foreign policy](https://en.wikipedia.org/wiki/Foreign_policy), or supply chain management.

In agriculture, AI has helped farmers identify areas that need irrigation, fertilization, pesticide treatments or increasing yield. Agronomists use AI to conduct research and development. AI has been used to predict the ripening time for crops such as tomatoes, monitor soil moisture, operate agricultural robots, conduct predictive analytics, classify livestock pig call emotions, automate greenhouses, detect diseases and pests, and save water.

Artificial intelligence is used in astronomy to analyze increasing amounts of available data and applications, mainly for "classification, regression, clustering, forecasting, generation, discovery, and the development of new scientific insights" for example for discovering exoplanets, forecasting solar activity, and distinguishing between signals and instrumental effects in gravitational wave astronomy. It could also be used for activities in space such as space exploration, including analysis of data from space missions, real-time science decisions of spacecraft, space debris avoidance, and more autonomous operation.

**1.5 Ethics**

AI has potential benefits and potential risks. AI may be able to advance science and find solutions for serious problems: [Demis Hassabis](https://en.wikipedia.org/wiki/Demis_Hassabis) of [Deep Mind](https://en.wikipedia.org/wiki/DeepMind) hopes to "solve intelligence, and then use that to solve everything else". However, as the use of AI has become widespread, several unintended consequences and risks have been identified. In-production systems can sometimes not factor ethics and bias into their AI training processes, especially when the AI algorithms are inherently unexplainable in deep learning.

### 1.51 Risks and harm

### 

#### 1.511 Privacy and copyright

Further information: [Information privacy](https://en.wikipedia.org/wiki/Information_privacy) and [Artificial intelligence and copyright](https://en.wikipedia.org/wiki/Artificial_intelligence_and_copyright)

Machine-learning algorithms require large amounts of data. The techniques used to acquire this data have raised concerns about [privacy](https://en.wikipedia.org/wiki/Privacy), [surveillance](https://en.wikipedia.org/wiki/Surveillance) and [copyright](https://en.wikipedia.org/wiki/Copyright).

Technology companies collect a wide range of data from their users, including online activity, geolocation data, video and audio. For example, in order to build [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition) algorithms, [Amazon](https://en.wikipedia.org/wiki/Amazon_(company)) has recorded millions of private conversations and allowed [temporary workers](https://en.wikipedia.org/wiki/Temporary_worker) to listen to and transcribe some of them. Opinions about this widespread [surveillance](https://en.wikipedia.org/wiki/Surveillance) range from those who see it as a [necessary evil](https://en.wikipedia.org/wiki/Necessary_evil) to those for whom it is clearly [unethical](https://en.wikipedia.org/wiki/Unethical) and a violation of the [right to privacy](https://en.wikipedia.org/wiki/Right_to_privacy).

AI developers argue that this is the only way to deliver valuable applications. and have developed several techniques that attempt to preserve privacy while still obtaining the data, such as [data aggregation](https://en.wikipedia.org/wiki/Data_aggregation), [de-identification](https://en.wikipedia.org/wiki/De-identification) and [differential privacy](https://en.wikipedia.org/wiki/Differential_privacy). Since 2016, some privacy experts, such as [Cynthia Dwork](https://en.wikipedia.org/wiki/Cynthia_Dwork), have begun to view privacy in terms of [fairness](https://en.wikipedia.org/wiki/Fairness_(machine_learning)). [Brian Christian](https://en.wikipedia.org/wiki/Brian_Christian) wrote that experts have pivoted "from the question of 'what they know' to the question of 'what they're doing with it'."

Generative AI is often trained on unlicensed copyrighted works, including in domains such as images or computer code; the output is then used under the rationale of "[fair use](https://en.wikipedia.org/wiki/Fair_use)". Website owners who do not wish to have their copyrighted content AI-indexed or 'scraped' can add code to their site if they do not want their website to be indexed by a search engine, which is currently available through certain services such as [OpenAI](https://en.wikipedia.org/wiki/OpenAI). Experts disagree about how well and under what circumstances this rationale will hold up in courts of law; relevant factors may include "the purpose and character of the use of the copyrighted work" and "the effect upon the potential market for the copyrighted work" In 2023, leading authors (including [John Grisham](https://en.wikipedia.org/wiki/John_Grisham) and [Jonathan Franzen](https://en.wikipedia.org/wiki/Jonathan_Franzen)) sued AI companies for using their work to train generative AI.

#### 1.512 Misinformation

[YouTube](https://en.wikipedia.org/wiki/YouTube), [Facebook](https://en.wikipedia.org/wiki/Facebook) and others use [recommender systems](https://en.wikipedia.org/wiki/Recommender_system) to guide users to more content. These AI programs were given the goal of [maximizing](https://en.wikipedia.org/wiki/Mathematical_optimization) user engagement (that is, the only goal was to keep people watching). The AI learned that users tended to choose [misinformation](https://en.wikipedia.org/wiki/Misinformation), [conspiracy theories](https://en.wikipedia.org/wiki/Conspiracy_theory), and extreme [partisan](https://en.wikipedia.org/wiki/Partisan_(politics)) content, and, to keep them watching, the AI recommended more of it. Users also tended to watch more content on the same subject, so the AI led people into [filter bubbles](https://en.wikipedia.org/wiki/Filter_bubbles) where they received multiple versions of the same misinformation. This convinced many users that the misinformation was true, and ultimately undermined trust in institutions, the media and the government. The AI program had correctly learned to maximize its goal, but the result was harmful to society. After the U.S. election in 2016, major technology companies took steps to mitigate the problem.

In 2022, [generative AI](https://en.wikipedia.org/wiki/Generative_AI) began to create images, audio, video and text that are indistinguishable from real photographs, recordings, films or human writing. It is possible for bad actors to use this technology to create massive amounts of misinformation or propaganda. AI pioneer [Geoffrey Hinton](https://en.wikipedia.org/wiki/Geoffrey_Hinton) expressed concern about AI enabling "authoritarian leaders to manipulate their electorates" on a large scale, among other risks.

#### 1.513 Algorithmic bias and fairness

Machine learning applications will be biased if they learn from biased data. The developers may not be aware that the bias exists. Bias can be introduced by the way [training data](https://en.wikipedia.org/wiki/Training_data) is selected and by the way a model is deployed. [Fairness](https://en.wikipedia.org/wiki/Fairness_(machine_learning)) in machine learning is the study of how to prevent the harm caused by algorithmic bias. It has become serious area of academic study within AI. Researchers have discovered it is not always possible to define "fairness" in a way that satisfies all stakeholders.

On June 28, 2015, [Google Photos](https://en.wikipedia.org/wiki/Google_Photos)'s new image labeling feature mistakenly identified Jacky Alcine and a friend as "gorillas" because they were black. The system was trained on a dataset that contained very few images of black people, a problem called "sample size disparity". Google "fixed" this problem by preventing the system from labelling anything as a "gorilla". Eight years later, in 2023, Google Photos still could not identify a gorilla, and neither could similar products from Apple, Facebook, Microsoft and Amazon.

A program can make biased decisions even if the data does not explicitly mention a problematic feature (such as "race" or "gender"). The feature will correlate with other features (like "address", "shopping history" or "first name"), and the program will make the same decisions based on these features as it would on "race" or "gender".Moritz Hardt said "the most robust fact in this research area is that fairness through blindness doesn't work."

Criticism of COMPAS highlighted that machine learning models are designed to make "predictions" that are only valid if we assume that the future will resemble the past. If they are trained on data that includes the results of racist decisions in the past, machine learning models must predict that racist decisions will be made in the future. If an application then uses these predictions as recommendations, some of these "recommendations" will likely be racist. Thus, machine learning is not well suited to help make decisions in areas where there is hope that the future will be better than the past. It is necessarily descriptive and not proscriptive.

Bias and unfairness may go undetected because the developers are overwhelmingly white and male: among AI engineers, about 4% are black and 20% are women.

At its 2022 [Conference on Fairness, Accountability, and Transparency](https://en.wikipedia.org/wiki/ACM_Conference_on_Fairness,_Accountability,_and_Transparency) (ACM FAccT 2022), the [Association for Computing Machinery](https://en.wikipedia.org/wiki/Association_for_Computing_Machinery), in Seoul, South Korea, presented and published findings that recommend that until AI and robotics systems are demonstrated to be free of bias mistakes, they are unsafe, and the use of self-learning neural networks trained on vast, unregulated sources of flawed internet data should be curtailed.

#### 1.514 Lack of transparency

[](https://en.wikipedia.org/wiki/File:HiPhi_Z,_IAA_Summit_2023,_Munich_(P1120237).jpg)

[Lidar](https://en.wikipedia.org/wiki/Lidar) testing vehicle for autonomous driving

Many AI systems are so complex that their designers cannot explain how they reach their decisions.  Particularly with [deep neural networks](https://en.wikipedia.org/wiki/Deep_neural_networks), in which there are a large amount of non-[linear](https://en.wikipedia.org/wiki/Linear) relationships between inputs and outputs. But some popular explainability techniques exist.

It is impossible to be certain that a program is operating correctly if no one knows how exactly it works. There have been many cases where a machine learning program passed rigorous tests, but nevertheless learned something different than what the programmers intended. For example, a system that could identify skin diseases better than medical professionals was found to actually have a strong tendency to classify images with a [ruler](https://en.wikipedia.org/wiki/Ruler) as "cancerous", because pictures of malignancies typically include a ruler to show the scale. Another machine learning system designed to help effectively allocate medical resources was found to classify patients with asthma as being at "low risk" of dying from pneumonia. Having asthma is actually a severe risk factor, but since the patients having asthma would usually get much more medical care, they were relatively unlikely to die according to the training data. The correlation between asthma and low risk of dying from pneumonia was real, but misleading.

People who have been harmed by an algorithm's decision have a right to an explanation.  Doctors, for example, are expected to clearly and completely explain to their colleagues the reasoning behind any decision they make. Early drafts of the European Union's [General Data Protection Regulation](https://en.wikipedia.org/wiki/General_Data_Protection_Regulation) in 2016 included an explicit statement that this right exists. Industry experts noted that this is an unsolved problem with no solution in sight. Regulators argued that nevertheless the harm is real: if the problem has no solution, the tools should not be used.

[DARPA](https://en.wikipedia.org/wiki/DARPA) established the [XAI](https://en.wikipedia.org/wiki/Explainable_Artificial_Intelligence) ("Explainable Artificial Intelligence") program in 2014 to try and solve these problems. There are several possible solutions to the transparency problem. SHAP tried to solve the transparency problems by visualising the contribution of each feature to the output. LIME can locally approximate a model with a simpler, interpretable model. [Multitask learning](https://en.wikipedia.org/wiki/Multitask_learning) provides a large number of outputs in addition to the target classification. These other outputs can help developers deduce what the network has learned. [Deconvolution](https://en.wikipedia.org/wiki/Deconvolution), [Deep Dream](https://en.wikipedia.org/wiki/DeepDream) and other [generative](https://en.wikipedia.org/wiki/Generative_AI) methods can allow developers to see what different layers of a deep network have learned and produce output that can suggest what the network is learning.

#### 1.515 Bad actors and weaponized AI

Artificial intelligence provides a number of tools that are useful to [bad actors](https://en.wikipedia.org/wiki/Bad_actor), such as [authoritarian governments](https://en.wikipedia.org/wiki/Authoritarian), [terrorists](https://en.wikipedia.org/wiki/Terrorist), [criminals](https://en.wikipedia.org/wiki/Criminals) or [rogue states](https://en.wikipedia.org/wiki/Rogue_states).

A lethal autonomous weapon is a machine that locates, selects and engages human targets without human supervision. Widely available AI tools can be used by bad actors to develop inexpensive autonomous weapons and, if produced at scale, they are potentially [weapons of mass destruction](https://en.wikipedia.org/wiki/Weapons_of_mass_destruction). Even when used in conventional warfare, it is unlikely that they will be unable to reliably choose targets and could potentially [kill an innocent person](https://en.wikipedia.org/wiki/Murder) In 2014, 30 nations (including China) supported a ban on autonomous weapons under the [United Nations](https://en.wikipedia.org/wiki/United_Nations)' [Convention on Certain Conventional Weapons](https://en.wikipedia.org/wiki/Convention_on_Certain_Conventional_Weapons), however the [United States](https://en.wikipedia.org/wiki/United_States) and others disagreed By 2015, over fifty countries were reported to be researching battlefield robots.

AI tools make it easier for [authoritarian governments](https://en.wikipedia.org/wiki/Authoritarian) to efficiently control their citizens in several ways. [Face](https://en.wikipedia.org/wiki/Facial_recognition_system) and [voice recognition](https://en.wikipedia.org/wiki/Speaker_recognition) allow widespread [surveillance](https://en.wikipedia.org/wiki/Surveillance). [Machine learning](https://en.wikipedia.org/wiki/Machine_learning), operating this data, can [classify](https://en.wikipedia.org/wiki/Classifier_(machine_learning)) potential enemies of the state and prevent them from hiding.  [Recommendation systems](https://en.wikipedia.org/wiki/Recommender_system) can precisely target [propaganda](https://en.wikipedia.org/wiki/Propaganda) and [misinformation](https://en.wikipedia.org/wiki/Misinformation) for maximum effect. [Deepfakes](https://en.wikipedia.org/wiki/Deepfakes) and [generative AI](https://en.wikipedia.org/wiki/Generative_AI) aid in producing misinformation. Advanced AI can make authoritarian [centralized decision making](https://en.wikipedia.org/wiki/Technocracy) more competitive than liberal and decentralized systems such as [markets](https://en.wikipedia.org/wiki/Market_(economics)). It lowers the cost and difficulty of [digital warfare](https://en.wikipedia.org/wiki/Digital_warfare) and [advanced spyware](https://en.wikipedia.org/wiki/Spyware). All these technologies have been available since 2020 or earlier—AI [facial recognition systems](https://en.wikipedia.org/wiki/Facial_recognition_system) are already being used for [mass surveillance](https://en.wikipedia.org/wiki/Mass_surveillance) in China.

There many other ways that AI is expected to help bad actors, some of which can not be foreseen. For example, machine-learning AI is able to design tens of thousands of toxic molecules in a matter of hours.

#### 1.516 Reliance on industry giants

Training AI systems requires an enormous amount of computing power. Usually only [Big Tech](https://en.wikipedia.org/wiki/Big_Tech) companies have the financial resources to make such investments. Smaller startups such as [Cohere](https://en.wikipedia.org/wiki/Cohere) and [OpenAI](https://en.wikipedia.org/wiki/OpenAI) end up buying access to [data centers](https://en.wikipedia.org/wiki/Data_centers) from [Google](https://en.wikipedia.org/wiki/Google) and [Microsoft](https://en.wikipedia.org/wiki/Microsoft) respectively

#### 1.517 Technological unemployment

Economists have frequently highlighted the risks of redundancies from AI, and speculated about unemployment if there is no adequate social policy for full employment.

In the past, technology has tended to increase rather than reduce total employment, but economists acknowledge that "we're in uncharted territory" with AI. A survey of economists showed disagreement about whether the increasing use of robots and AI will cause a substantial increase in long-term [unemployment](https://en.wikipedia.org/wiki/Unemployment), but they generally agree that it could be a net benefit if [productivity](https://en.wikipedia.org/wiki/Productivity) gains are [redistributed](https://en.wikipedia.org/wiki/Redistribution_of_income_and_wealth). Risk estimates vary; for example, in the 2010s, Michael Osborne and [Carl Benedikt Frey](https://en.wikipedia.org/wiki/Carl_Benedikt_Frey) estimated 47% of U.S. jobs are at "high risk" of potential automation, while an OECD report classified only 9% of U.S. jobs as "high risk". The methodology of speculating about future employment levels has been criticised as lacking evidential foundation, and for implying that technology, rather than social policy, creates unemployment, as opposed to redundancies. In April 2023, it was reported that 70% of the jobs for Chinese video game illustrators had been eliminated by generative artificial intelligence.

Unlike previous waves of automation, many middle-class jobs may be eliminated by artificial intelligence; [The Economist](https://en.wikipedia.org/wiki/The_Economist) stated in 2015 that "the worry that AI could do to white-collar jobs what steam power did to blue-collar ones during the Industrial Revolution" is "worth taking seriously". Jobs at extreme risk range from [paralegals](https://en.wikipedia.org/wiki/Paralegal) to fast food cooks, while job demand is likely to increase for care-related professions ranging from personal healthcare to the clergy.

From the early days of the development of artificial intelligence, there have been arguments, for example, those put forward by [Joseph Weizenbaum](https://en.wikipedia.org/wiki/Joseph_Weizenbaum), about whether tasks that can be done by computers actually should be done by them, given the difference between computers and humans, and between quantitative calculation and qualitative, value-based judgement.

#### 1.518 Existential risk

It has been argued AI will become so powerful that humanity may irreversibly lose control of it. This could, as physicist [Stephen Hawking](https://en.wikipedia.org/wiki/Stephen_Hawking) stated, "[spell the end of the human race](https://en.wikipedia.org/wiki/Global_catastrophic_risk)". This scenario has been common in science fiction, when a computer or robot suddenly develops a human-like "self-awareness" (or "sentience" or "consciousness") and becomes a malevolent character.[[p]](https://en.wikipedia.org/wiki/Artificial_intelligence#cite_note-228) These sci-fi scenarios are misleading in several ways.

First, AI does not require human-like "[sentience](https://en.wikipedia.org/wiki/Sentience)" to be an existential risk. Modern AI programs are given specific goals and use learning and intelligence to achieve them. Philosopher [Nick Bostrom](https://en.wikipedia.org/wiki/Nick_Bostrom) argued that if one gives almost any goal to a sufficiently powerful AI, it may choose to destroy humanity to achieve it (he used the example of a [paperclip factory manager](https://en.wikipedia.org/wiki/Instrumental_convergence#Paperclip_maximizer)). [Stuart Russell](https://en.wikipedia.org/wiki/Stuart_J._Russell) gives the example of household robot that tries to find a way to kill its owner to prevent it from being unplugged, reasoning that "you can't fetch the coffee if you're dead." In order to be safe for humanity, a [superintelligence](https://en.wikipedia.org/wiki/Superintelligence) would have to be genuinely [aligned](https://en.wikipedia.org/wiki/AI_alignment) with humanity's morality and values so that it is "fundamentally on our side".

Second, [Yuval Noah Harari](https://en.wikipedia.org/wiki/Yuval_Noah_Harari) argues that AI does not require a robot body or physical control to pose an existential risk. The essential parts of civilization are not physical. Things like [ideologies](https://en.wikipedia.org/wiki/Ideology), [law](https://en.wikipedia.org/wiki/Law), [government](https://en.wikipedia.org/wiki/Government), [money](https://en.wikipedia.org/wiki/Money) and the [economy](https://en.wikipedia.org/wiki/Economy) are made of [language](https://en.wikipedia.org/wiki/Language); they exist because there are stories that billions of people believe. The current prevalence of [misinformation](https://en.wikipedia.org/wiki/Misinformation) suggests that an AI could use language to convince people to believe anything, even to take actions that are destructive.

The opinions amongst experts and industry insiders are mixed, with sizable fractions both concerned and unconcerned by risk from eventual superintelligent AI. Personalities such as [Stephen Hawking](https://en.wikipedia.org/wiki/Stephen_Hawking), [Bill Gates](https://en.wikipedia.org/wiki/Bill_Gates), and [Elon Musk](https://en.wikipedia.org/wiki/Elon_Musk) have expressed concern about existential risk from AI. AI pioneers including [Fei-Fei Li](https://en.wikipedia.org/wiki/Fei-Fei_Li), [Geoffrey Hinton](https://en.wikipedia.org/wiki/Geoffrey_Hinton), [Yoshua Bengio](https://en.wikipedia.org/wiki/Yoshua_Bengio), [Cynthia Breazeal](https://en.wikipedia.org/wiki/Cynthia_Breazeal), [Rana el Kaliouby](https://en.wikipedia.org/wiki/Rana_el_Kaliouby), [Demis Hassabis](https://en.wikipedia.org/wiki/Demis_Hassabis), [Joy Buolamwini](https://en.wikipedia.org/wiki/Joy_Buolamwini), and [Sam Altman](https://en.wikipedia.org/wiki/Sam_Altman) have expressed concerns about the risks of AI. In 2023, many leading AI experts issued the joint statement that "Mitigating the risk of extinction from AI should be a global priority alongside other societal-scale risks such as pandemics and nuclear war".

Other researchers, however, spoke in favor of a less dystopian view. AI pioneer [Juergen Schmidhuber](https://en.wikipedia.org/wiki/Juergen_Schmidhuber) did not sign the joint statement, emphasising that in 95% of all cases, AI research is about making "human lives longer and healthier and easier."  While the tools that are now being used to improve lives can also be used by bad actors, "they can also be used against the bad actors." [Andrew Ng](https://en.wikipedia.org/wiki/Andrew_Ng) also argued that "it's a mistake to fall for the doomsday hype on AI—and that regulators who do will only benefit vested interests." [Yann LeCun](https://en.wikipedia.org/wiki/Yann_LeCun) "scoffs at his peers' dystopian scenarios of supercharged misinformation and even, eventually, human extinction." In the early 2010s, experts argued that the risks are too distant in the future to warrant research or that humans will be valuable from the perspective of a superintelligent machine. However, after 2016, the study of current and future risks and possible solutions became a serious area of research.

### 1.52 Ethical machines and alignment

Friendly AI are machines that have been designed from the beginning to minimize risks and to make choices that benefit humans. [Eliezer Yudkowsky](https://en.wikipedia.org/wiki/Eliezer_Yudkowsky), who coined the term, argues that developing friendly AI should be a higher research priority: it may require a large investment and it must be completed before AI becomes an existential risk.

Machines with intelligence have the potential to use their intelligence to make ethical decisions. The field of machine ethics provides machines with ethical principles and procedures for resolving ethical dilemmas. The field of machine ethics is also called computational morality,  and was founded at an [AAAI](https://en.wikipedia.org/wiki/AAAI) symposium in 2005.

Other approaches include [Wendell Wallach](https://en.wikipedia.org/wiki/Wendell_Wallach)'s "artificial moral agents" and [Stuart J. Russell](https://en.wikipedia.org/wiki/Stuart_J._Russell)'s [three principles](https://en.wikipedia.org/wiki/Human_Compatible#Russell's_three_principles) for developing provably beneficial machines.

### 1.53 Open Source

Active organizations in the AI open-source community include [Hugging Face](https://en.wikipedia.org/wiki/Hugging_Face), [Google](https://en.wikipedia.org/wiki/Google), [Eleuther AI](https://en.wikipedia.org/wiki/EleutherAI" \o "EleutherAI) and [Meta](https://en.wikipedia.org/wiki/Meta_Platforms). Various AI models, such as [Llama 2](https://en.wikipedia.org/wiki/LLaMA), [Mistral](https://en.wikipedia.org/wiki/Mistral_AI) or [Stable Diffusion](https://en.wikipedia.org/wiki/Stable_Diffusion), have been made open-weight, meaning that their architecture and trained parameters (the "weights") are publicly available. Open-weight models can be freely [fine-tuned](https://en.wikipedia.org/wiki/Fine-tuning_(deep_learning)), which allows companies to specialize them with their own data and for their own use-case. Open-weight models are useful for research and innovation, but can also be misused. Since they can be fine-tuned, any built-in security measure, such as objecting to harmful requests, can be trained away until it becomes ineffective. Some researchers warn that future AI models may develop dangerous capabilities (such as the potential to drastically facilitate [bioterrorism](https://en.wikipedia.org/wiki/Bioterrorism)), and that once released on the Internet, they can't be deleted everywhere if needed. They recommend pre-release audits and cost-benefit analyses.

### 1.54 Frameworks

Artificial Intelligence projects can have their ethical permissibility tested while designing, developing, and implementing an AI system. An AI framework such as the Care and Act Framework containing the SUM values—developed by the [Alan Turing Institute](https://en.wikipedia.org/wiki/Alan_Turing_Institute) tests projects in four main areas:

* RESPECT the dignity of individual people
* CONNECT with other people sincerely, openly and inclusively
* CARE for the wellbeing of everyone
* PROTECT social values, justice and the public interest

Other developments in ethical frameworks include those decided upon during the [Asilomar Conference](https://en.wikipedia.org/wiki/Asilomar_Conference_on_Beneficial_AI), the Montreal Declaration for Responsible AI, and the IEEE's Ethics of Autonomous Systems initiative, among others;  however, these principles do not go without their criticisms, especially regards to the people chosen contributes to these frameworks.

Promotion of the wellbeing of the people and communities that these technologies affect requires consideration of the social and ethical implications at all stages of AI system design, development and implementation, and collaboration between job roles such as data scientists, product managers, data engineers, domain experts, and delivery managers.

**CHAPTER 2**

**SYSTEM ANALYSIS AND DESCRIPTION**

The development of AI systems for identifying inappropriate language and hate speech involves a series of steps, from data collection and preprocessing to model training and evaluation. This section provides a detailed analysis and description of the key components involved in this process. The development of AI systems for identifying inappropriate language and hate speech is a complex and multidisciplinary process that requires careful consideration of data, algorithms, and ethical considerations. By addressing these challenges and leveraging the latest advancements in AI research, we can develop more effective and robust systems for combating online toxicity.AI systems for detecting inappropriate language and hate speech rely on large datasets containing examples of such content. These datasets are often collected from social media platforms, online forums, and other sources where inappropriate language and hate speech are prevalent. Before training a model, the data must be preprocessed to remove noise, such as irrelevant characters or formatting issues. This step also involves tokenization, where the text is split into individual words or tokens, and normalization, where words are converted to their base form (e.g., "running" to "run").

* 1. **EXISTING SYSTEM**

The input text is preprocessed to remove irrelevant information, such as punctuation and stop words, and to normalize the text (e.g., converting all letters to lowercase). Features are extracted from the preprocessed text, such as word frequencies, n-grams (sequences of n words), and syntactic features. Various machine learning models are used to classify text as either inappropriate/hateful or not. Common models include linear classifiers (e.g., logistic regression, SVM), tree-based models (e.g., decision trees, random forests), and neural networks (e.g., LSTM, CNN). Once trained, the models can be deployed to classify new text inputs. They can be integrated into platforms such as social media sites, online forums, and chat applications to automatically flag or filter out inappropriate content. While these systems have shown promising results, they are not without limitations. They may struggle with context-dependent language, subtle forms of hate speech, and evolving language trends. Ongoing research is focused on improving the accuracy and robustness of these systems, as well as addressing ethical concerns related to censorship and bias.

The input text undergoes preprocessing to remove irrelevant information such as punctuation and stop words. Additionally, text normalization techniques are applied, such as converting all letters to lowercase. These preprocessing steps help in standardizing the input text for further analysis.

After preprocessing, features are extracted from the text data. These features include word frequencies, n-grams (sequences of n words), and syntactic features. Extracting meaningful features from the text data is crucial for training machine learning models to classify text effectively.

Various machine learning models are utilized for classifying text as either inappropriate/hateful or not. Commonly used models include linear classifiers (e.g., logistic regression, SVM), tree-based models (e.g., decision trees, random forests), and neural networks (e.g., LSTM, CNN). These models are trained using labeled data to learn the patterns and characteristics of inappropriate language and hate speech.

Once trained, the models are deployed to classify new text inputs. They can be integrated into platforms such as social media sites, online forums, and chat applications to automatically flag or filter out inappropriate content. This deployment allows for real-time identification and moderation of inappropriate language and hate speech.

While the existing systems have shown promising results, they are not without limitations. Some of the challenges include:

- Struggle with context-dependent language: AI models may have difficulty understanding the context in which language is used, leading to misclassification.

- Subtle forms of hate speech: Detecting subtle forms of hate speech can be challenging for AI models, as they may require deeper semantic understanding.

- Evolving language trends: Language is constantly evolving, and AI models may struggle to keep up with new slang, cultural references, and modes of expression.

- Ethical concerns: There are ethical concerns related to censorship and bias, as AI models may inadvertently perpetuate biases present in the training data.

Ongoing research is focused on improving the accuracy and robustness of these systems. This includes developing more advanced natural language processing techniques, enhancing model interpretability, and addressing ethical concerns related to censorship and bias. By continuously advancing the state-of-the-art in AI for identifying inappropriate language and hate speech, researchers aim to create more effective and fair content moderation systems.

In addition to the challenges mentioned, there are further complexities in the existing systems for identifying inappropriate language and hate speech using AI. One significant challenge is the dynamic nature of online communication and the rapid evolution of language trends. Slang terms, cultural references, and new modes of expression constantly emerge, making it challenging for AI models to keep pace and accurately classify language.

Moreover, the detection of hate speech often involves nuanced understanding of socio-political contexts and cultural sensitivities. AI models may struggle to interpret subtle nuances and contextual cues, leading to potential misclassifications. For instance, a statement that may appear innocuous in one cultural context could be highly offensive in another, highlighting the need for more sophisticated algorithms capable of cross-cultural sensitivity.

Furthermore, the ethical implications of content moderation using AI are complex and multifaceted. There are concerns regarding the potential for algorithmic bias, where AI models may inadvertently discriminate against certain groups or amplify existing societal biases present in the training data. Additionally, the automated censorship of content raises questions about freedom of speech and the appropriate balance between content moderation and user autonomy.

As AI systems for identifying inappropriate language and hate speech become more ubiquitous, there is a pressing need for interdisciplinary collaboration between technologists, social scientists, ethicists, and policymakers. Collaborative efforts can help address the ethical and societal implications of AI-driven content moderation while advancing the development of more accurate, transparent, and fair systems.

Ultimately, the goal is to create AI systems that not only effectively identify and mitigate inappropriate language and hate speech but also uphold principles of fairness, accountability, and respect for diverse perspectives. This requires ongoing research, innovation, and dialogue to navigate the complex challenges at the intersection of technology and society.

**2.11 DRAWBACKS**

AI models can inherit biases present in the training data, leading to unfair or discriminatory outcomes. For example, if the training data contains biased labels or reflects societal biases, the model may learn and perpetuate these biases. AI models may struggle to understand the context in which language is used, leading to errors in classification. For example, a model may incorrectly flag a statement as hate speech when it is used sarcastically or in a non-offensive context. Language is constantly evolving, and AI models may struggle to keep up with new slang, cultural references, and modes of expression. This can lead to outdated or ineffective models over time. AI models may produce false positives, flagging content as inappropriate or hateful when it is not. This can lead to censorship and frustration among users. Some AI models used for identifying inappropriate language and hate speech are complex and difficult to interpret, making it challenging to understand why certain decisions are made.

* 1. **PROPOSED SYSTEM**

The proposed system will employ techniques to mitigate bias in the training data and in the AI models themselves. This may include using balanced datasets, bias-aware learning algorithms, and post-processing techniques to reduce bias in predictions. The proposed system will incorporate advanced natural language processing (NLP) techniques to better understand the context in which language is used. This will help reduce false positives by taking into account the nuanced meanings of words and phrases. The proposed system will be designed to continuously learn from new data and user feedback. This will help the system adapt to evolving language trends and user behaviour, improving its accuracy over time. The proposed system will take into account ethical considerations, such as ensuring that the system does not unfairly target specific groups or censor legitimate speech.

The proposed system for identifying inappropriate language and hate speech using AI aims to revolutionize content moderation by addressing the limitations of existing systems. At its core, the system focuses on mitigating biases, leveraging advanced NLP techniques, enabling continuous learning, and upholding ethical considerations.

Bias mitigation techniques play a crucial role in ensuring fair and equitable outcomes in content moderation. By employing balanced datasets, bias-aware learning algorithms, and post-processing techniques, the proposed system aims to reduce biases present in both the training data and AI models.

Advanced NLP techniques are key to understanding the nuanced context of language usage. By analyzing the subtle meanings of words and phrases, the system can significantly reduce false positives in identifying inappropriate language and hate speech. This leads to improved accuracy and effectiveness in content moderation.

Continuous learning and adaptation are fundamental aspects of the proposed system. By continuously learning from new data and user feedback, the system can evolve and adapt to changing language trends and user behavior. This adaptive approach ensures that the system remains effective and relevant over time.

Ethical considerations are paramount in the design and implementation of the proposed system. Upholding principles of fairness, transparency, and accountability is essential to ensure that the system does not unfairly target specific groups or censor legitimate speech.

The advantages of the proposed system are manifold. By reducing biases, improving context understanding, enabling continuous learning, and prioritizing ethics, the system offers significant improvements over existing systems. It leads to more fair and equitable outcomes in content moderation, enhances accuracy in identifying inappropriate language and hate speech, and ensures adaptability to evolving language trends and user behavior.

Overall, the proposed system represents a groundbreaking advancement in AI-based content moderation. It holds the potential to transform how we approach the identification and mitigation of inappropriate language and hate speech, paving the way for a safer and more inclusive online environment.

**2.21 ADVANTAGES**

The proposed system for identifying inappropriate language and hate speech using AI offers several advantages over existing systems, By employing bias mitigation techniques, the proposed system can reduce the impact of biases present in the training data and AI models. This can lead to more fair and equitable outcomes in content moderation. The proposed system's use of advanced NLP techniques enables better understanding of the context in which language is used. This can help reduce false positives and improve the accuracy of identifying inappropriate language and hate speech. The proposed system's ability to continuously learn from new data and user feedback allows it to adapt to evolving language trends and user behaviour. This can lead to improved accuracy and effectiveness over time.

**2.3 FEASIBILITY STUDY**

Detecting inappropriate language, including hate speech, through AI techniques such as BERT presents a promising avenue for mitigating online toxicity. The prevalence and impact of hate speech online underscore the urgency of employing automated systems for content moderation and ensuring online safety. BERT, with its contextual understanding capabilities, stands out as a powerful tool in this endeavor. Additionally, other AI techniques like machine learning classifiers and rule-based systems offer complementary approaches to identifying inappropriate language.

Training BERT for hate speech detection involves fine-tuning pre-trained models on annotated datasets, highlighting the importance of quality data for effective model performance. BERT's ability to capture contextual information enables it to discern nuances in language, distinguishing between harmful and non-harmful content. Furthermore, exploring alternative AI techniques provides flexibility in addressing different aspects of inappropriate language detection, each with its own set of strengths and limitations.

Considering the feasibility of employing BERT and other AI techniques, factors such as model availability, computational resources, and real-world performance metrics play crucial roles. While BERT's pre-trained models are readily accessible, computational requirements for fine-tuning and inference need consideration. Real-world performance metrics, including accuracy, precision, and recall, guide the assessment of AI models' effectiveness in hate speech detection tasks.

Case studies and examples demonstrating the efficacy of BERT and other AI techniques in identifying inappropriate language provide empirical evidence of their feasibility and impact. These implementations showcase successful applications of AI-driven content moderation, illustrating tangible outcomes in enhancing online safety and fostering healthier digital environments.

However, challenges such as linguistic variations and biases in AI models pose ongoing considerations in hate speech detection. Future research directions should address these challenges and explore innovative approaches to improving the effectiveness and fairness of AI-driven content moderation systems. In conclusion, leveraging AI for identifying inappropriate language, including hate speech, holds immense potential in creating safer online environments, reinforcing the need for continued exploration and development in this critical domain.

**2.4 MODULES DESCRIPTION**

Text Preprocessing

Feature Extraction

Model Training

Model Evaluation

Model Deployment

**2.41 Text Preprocessing**

Text preprocessing in the context of identifying inappropriate language and hate speech using AI involves several steps to clean and prepare the text data for analysis, This ensures that words are treated the same regardless of their casing. Split the text into individual words or tokens. This step is essential for further analysis as it breaks down the text into manageable units. Remove common words that do not carry much meaning, such as "the," "is," "and," etc. These words can be ignored in the analysis as they do not contribute much to identifying inappropriate language or hate speech. Correct any spelling mistakes in the text. This is important for ensuring that words are correctly interpreted during analysis.

**2.42 Feature Extraction**

Feature extraction in identifying inappropriate language and hate speech using AI involves transforming the preprocessed text data into a format that can be used by machine learning models. The goal is to extract meaningful features that capture the essence of the text and can help the model differentiate between different types of language.

**2.43 Model Training**

Model training in identifying inappropriate language and hate speech using AI involves training a machine learning model to classify text as either inappropriate or not inappropriate (i.e., hate speech or not hate speech).

**2.44 Model Evaluation**

Model evaluation in identifying inappropriate language and hate speech using AI involves assessing the performance of a trained model on a separate dataset that was not used for training. The goal is to measure how well the model generalizes to new, unseen data.

**2.45 Model Deployment**

Model deployment in the context of identifying inappropriate language and hate speech using AI refers to the process of making the trained model available for use in a production environment. Model deployment in the context of identifying inappropriate language and hate speech using AI refers to the process of making the trained model available for use in a production environment.

* 1. **ANAMOLY DETECTION**

Anomaly detection is a crucial technique in identifying inappropriate language and hate speech using AI. Inappropriate language and hate speech are often characterized by deviations from normal language patterns, making them suitable candidates for anomaly detection methods. This paper explores the use of anomaly detection techniques in identifying inappropriate language and hate speech, highlighting their benefits and challenges. Anomaly detection is the process of identifying patterns in data that do not conform to expected behaviour. In the context of identifying inappropriate language and hate speech, anomalies refer to text that deviates significantly from normal language patterns. Anomaly detection techniques aim to identify such anomalies by analysing various features of the text data. Anomaly detection models are trained on a dataset containing examples of normal language patterns. The model learns to distinguish between normal and anomalous patterns in the data. The performance of the model is evaluated using metrics such as precision, recall, and F1-score, which measure the model's ability to detect anomalies while minimizing false positives. Anomaly detection is a valuable technique for identifying inappropriate language and hate speech, as it can detect deviations from normal language patterns that may not be captured by traditional classification models. By leveraging the latest advancements in anomaly detection and addressing key challenges, we can develop more effective and robust models for combating harmful content online.

* 1. **MODEL EVALUATION AND VALIDATION**

Model evaluation and validation are essential steps in the development of AI systems for identifying inappropriate language and hate speech. These steps ensure that the models are accurate, reliable, and generalizable to new data. The evaluation process involves testing the models on labeled datasets and measuring their performance using various metrics. Validation, on the other hand, involves assessing the models' performance on unseen data to ensure that they can generalize well. Cross-validation is a technique used to evaluate the performance of AI models by splitting the dataset into multiple subsets, training the model on some subsets, and testing it on the remaining subset. This process is repeated multiple times, with each subset used for testing once. Cross-validation helps ensure that the model's performance is not overly dependent on the particular subset used for testing. Hyperparameter tuning is the process of optimizing the hyperparameters of the AI model to improve its performance. Hyperparameters are parameters that are not learned by the model but are set before training. Examples of hyperparameters include the learning rate, the number of hidden layers in a neural network, and the size of the word embeddings. Hyperparameter tuning can be done using techniques such as grid search, random search, or Bayesian optimization. Validation is the process of assessing the performance of AI models on unseen data to ensure that they can generalize well. This involves splitting the dataset into training, validation, and test sets. The model is trained on the training set, tuned on the validation set, and tested on the test set. Validation helps ensure that the model's performance is not overly optimistic due to overfitting.

**2.7 SCALABILITY AND DEPLOYMENT**

Scalability and deployment are crucial considerations in the development of AI systems for identifying inappropriate language and hate speech. These systems must be able to handle large volumes of data and be deployed in real-time to effectively combat harmful content online. This paper explores the challenges and strategies for achieving scalability and deployment in AI systems for identifying inappropriate language and hate speech. Several companies and organizations have successfully deployed AI systems for identifying inappropriate language and hate speech. For example, Twitter uses AI algorithms to detect and remove hateful content from its platform, while Facebook uses AI to detect and remove fake accounts and misinformation. Scalability and deployment are critical considerations in the development of AI systems for identifying inappropriate language and hate speech. By employing scalable architectures and deployment strategies, developers can create robust and effective AI systems that can handle large volumes of data and be deployed in real-time to combat harmful content online. Containerization platforms, such as Docker or Kubernetes, can be used to package AI models and their dependencies into lightweight, portable containers. This allows for easy deployment and scaling of AI systems. CI/CD pipelines can be used to automate the deployment process, ensuring that updates and changes to the AI system are deployed quickly and efficiently. Deployed systems should be monitored and logged to detect and troubleshoot issues. This requires implementing monitoring tools and logging mechanisms to track the system's performance and identify potential issues. Deployment refers to the process of making an AI system available for use in a production environment. Deploying AI systems for identifying inappropriate language and hate speech poses several challenges, AI systems need to be integrated with existing systems, such as social media platforms or content management systems, to effectively identify and mitigate inappropriate language and hate speech. Deployed systems must be able to scale to handle increasing amounts of data and traffic. This requires careful design and planning to ensure that the system can handle the workload efficiently. Deployed systems must be secure to protect against malicious attacks and unauthorized access. This requires implementing robust security measures, such as encryption and access control.

**2.8 ETHICAL CONSIDERATIONS AND PRIVACY**

AI models can inherit biases from training data, leading to unfair treatment of certain groups. It's crucial to address bias by ensuring diverse and representative training datasets. Balancing the need to combat hate speech with the right to free expression is challenging. AI systems must be designed to respect legal and ethical principles of free speech. Users should be informed when AI is used for content moderation. There should be clear mechanisms for appealing decisions and holding AI systems accountable for mistakes.

AI must be sensitive to cultural nuances to avoid misinterpretation of language and expressions that may be acceptable in one culture but offensive in another. AI systems require large amounts of data to function effectively, raising concerns about the collection of sensitive information without consent. Safeguarding user data is paramount to protect against data breaches and misuse by malicious actors. Users should be aware of and consent to the collection and use of their data for content moderation purposes. To protect user privacy, data should be anonymized whenever possible, ensuring that individuals cannot be identified from the data used by AI systems. Adherence to data protection regulations, such as the GDPR, is essential to ensure that user data is collected, processed, and stored legally and ethically. While AI can play a valuable role in identifying inappropriate language and hate speech, it is essential to address the ethical and privacy considerations outlined above to ensure that these systems are deployed responsibly and respect user rights.

**2.9 CONTINUOUS IMPROVEMENT AND ADAPTATION**

Continuous improvement and adaptation are essential for AI systems that aim to identify inappropriate language and hate speech. Implement a feedback mechanism where users can report misclassifications or provide feedback on the AI's decisions. This feedback can be used to improve the AI's accuracy and reduce false positives. Incorporate human moderators into the process to review and validate the AI's decisions. This can help improve the AI's performance and ensure that nuanced or context-dependent content is correctly identified. Use active learning techniques to select the most informative examples for human review, reducing the amount of human effort required to improve the AI's performance. Regularly retrain the AI model using updated datasets to keep up with evolving language and new forms of hate speech. Monitor changes in language use and societal norms to adapt the AI's algorithms and models to new contexts and emerging forms of inappropriate language and hate speech.

Collaborate with other organizations and researchers working on similar problems to share knowledge and best practices for improving AI-based content moderation. Ensure that improvements and adaptations to the AI system are made in a way that upholds ethical standards, such as fairness, transparency, and user privacy. By implementing these strategies, AI systems can continuously improve their ability to identify inappropriate language and hate speech while adapting to new challenges and contexts.

**CHAPTER 3**

**REQUIEMENT ANALYSIS**

**3.1 Software Requirements**

For developing the application the following are the Software Requirements:

1.Python

2.VS code

* 1. **Operating Systems supported**

Windows 10 64 bit OS

**3.3 Debugger and Emulator**

Any Browser (Particularly Chrome)

* 1. **Functional Requirements**

Setting up a Virtual environment

**3.4 Library Requirements**

**3.41 PyTorch**

* **Purpose**: PyTorch is a popular deep learning framework that provides dynamic computation graphs, making it well-suited for research and experimentation.
* **Applications**: It’s widely used for tasks like image recognition, natural language understanding, and reinforcement learning.

**3.42 TensorFlow**

* **Purpose**: TensorFlow is an open-source machine learning library developed by Google. It’s versatile and widely adopted in both research and production.
* **Applications**: Used for various tasks, including computer vision, speech recognition, and recommendation systems.

**3.43 Scikit-learn**

* **Purpose**: Scikit-learn is a powerful Python library for traditional machine learning tasks.
* **Applications**: It covers a wide range of tasks, including classification, regression, clustering, and dimensionality reduction.

**3.44 SpaCy**

* **Purpose:** SpaCy is a fast and efficient NLP library designed for production use. Its primary purpose is to process and analyze natural language text.
* **Applications:** SpaCy can identify entities such as names of people, organizations, locations, and dates in text

**3.45 NLTK**

* **Purpose**: NLTK is a comprehensive library for NLP research and education. It provides tools and resources for linguistic data analysis and modeling.
* **Applications**: NLTK offers tools for tokenization, stemming, and removing stopwords. It’s commonly used for cleaning and preparing text data.

**3.46 JSON**

* **Purpose**: JSON is used to store and structure the data required for hate speech detection, such as labeled datasets that include examples of text and their corresponding classifications.
* **Applications**: JSON files often represent the training data for hate speech detection models, with each entry containing the text to be analyzed and its label.

**3.47 Flask**

* **Purpose:** Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.
* **Applications:** Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier.

**CHAPTER 4**

**SOFTWARE DESIGN**

**4.1 Introduction**

In 2019, even Google has announced that it is using BERT in its search, supposedly the “biggest leap forward” it did in understanding search in the past five years. That is a huge testament to come from Google. About Search! That’s just how significant BERT is. It is worth mentioning, last year, Prabhakar Raghavan, Senior Vice President at Google, announced the launch of a new AI model called [Multitask Unified Model (MUM)](https://blog.google/products/search/introducing-mum/) at the [Google I/O](https://io.google/2021/session/88b34a4e-6170-4f18-a321-4260fb559e60/?lng=en) event. While this new model runs on the T5 framework, which is similar to BERT, [MUM is superior to BERT by 1000 times](https://blog.google/products/search/introducing-mum/) and is the future of AI behind google search engines.

**4.2 What is BERT?**

BERT is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google.[1][2] In 2019, Google announced that it had begun leveraging BERT in its search engine, and by late 2020 it was using BERT in almost every English-language query.

BERT is at its core a [transformer](https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)) language model with a variable number of encoder layers and self-attention heads. The architecture is "almost identical" to the original transformer implementation in [Vaswani et al. (2017).](https://arxiv.org/abs/1706.03762)

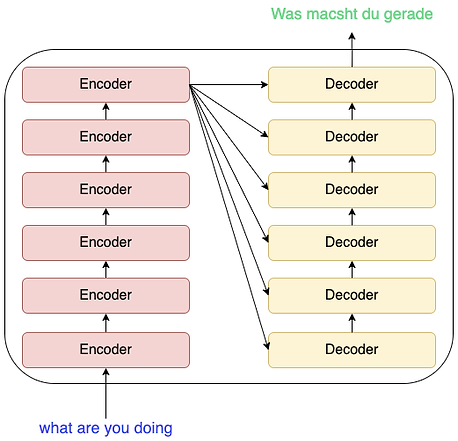
BERT makes use of [Transformer](https://jalammar.github.io/illustrated-transformer/), an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT’s goal is to generate a language model, only the encoder mechanism of the Transformer is necessary. The detailed workings of the Transformer are described in a [paper](https://arxiv.org/pdf/1706.03762.pdf) by Google.

As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it’s non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).

The bidirectionality of a model is important for truly understanding the meaning of a language. Let’s see an example to illustrate this. There are two sentences in this example and both of them involve the word “bank”:

If we try to predict the nature of the word “bank” by only taking either the left or the right context, then we will be making an error in at least one of the two given examples. One way to deal with this is to consider both the left and the right context before making a prediction. That’s exactly what BERT does! We will see later in the article how this is achieved.

The diagram below is a high-level description of the Transformer encoder and decoder. The input is a sequence of tokens, which are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors of size H, in which each vector corresponds to an input token with the same index.

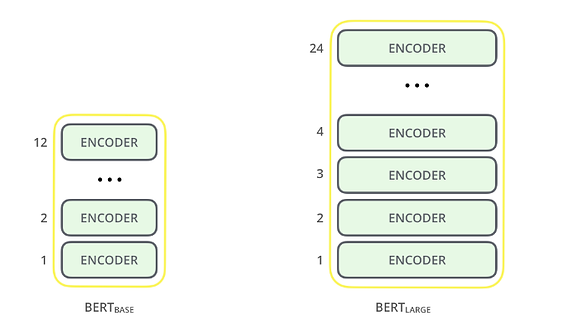


The above figure is a high-level diagram of a Transformer. [Credits](https://towardsdatascience.com/the-transformer-a-quick-run-through-ce9b21b4f3ed)The stack of Encoders is basically BERT, and if we just stack Decoders that are nothing but GPT(Generative Pre-trained Transformer), however, the BERT Transformer uses bidirectional self-attention, while the GPT Transformer uses constrained self-attention where every token can only attend to context to its left.

When training language models, there is a challenge of defining a prediction goal. Many models predict the next word in a sequence, a directional approach that inherently limits context learning. To overcome this challenge,

We currently have two variants available for BERT:

* **BERT Base**: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters
* **BERT Large**: 24 layers (transformer blocks), 16 attention heads, and, 340 million parameters



Both BERT model sizes have a large number of encoder layers (which the paper calls Transformer Blocks) – twelve for the Base version, and twenty-four for the Large version. These also have larger feedforward networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively) than the default configuration in the reference implementation of the Transformer in the initial paper (6 encoder layers, 512 hidden units, and 8 attention heads).

**4.3 How BERT work?**

There are two steps in the BERT framework: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data over different pre-training tasks. For finetuning, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks. Each downstream task has separate fine-tuned models, even though they are initialized with the same pre-trained parameters.

BERT pre-training uses two training strategies:

**4.31 Masked Language Modeling (MLM)**

BERT is designed as a deeplybi-directional model. The network effectively captures information from both the right and left context of a token from the first layer itself and all the way through to the last layer.

Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. In technical terms, the prediction of the output words requires:

1. Adding a classification layer on top of the encoder output.
2. Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.
3. Calculating the probability of each word in the vocabulary with softmax.

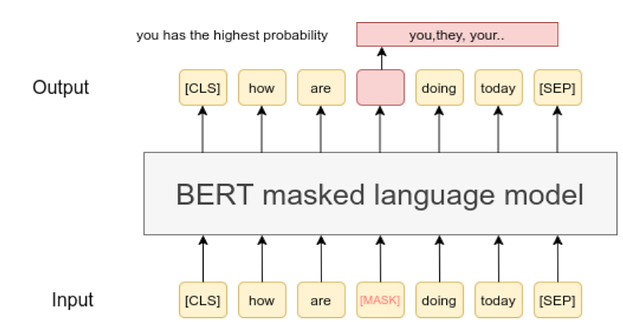
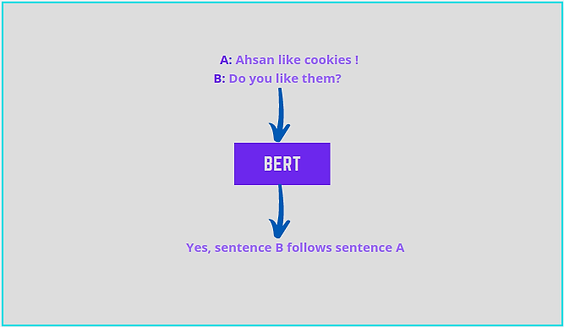


FIG Masked Language Modeling (MLM)

The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words. As a consequence, the model converges slower than directional models, a characteristic that is offset by its increased context-awareness.

**4.32 Next Sentence Prediction (NSP)**

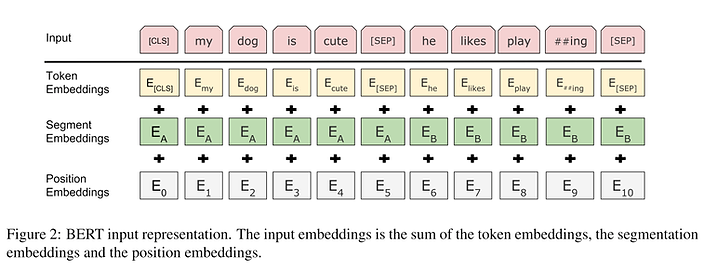
In the BERT training process, the model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. During training, 50% of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while in the other 50% a random sentence from the corpus is chosen as the second sentence. The assumption is that the random sentence will be disconnected from the first sentence.



Next Sentence Prediction(NSP). [Credits](https://zawster.wordpress.com/2020/08/24/a-beginners-guide-to-bert-architecture/)

To help the model distinguish between the two sentences in training, the input is processed in the following way before entering the model:

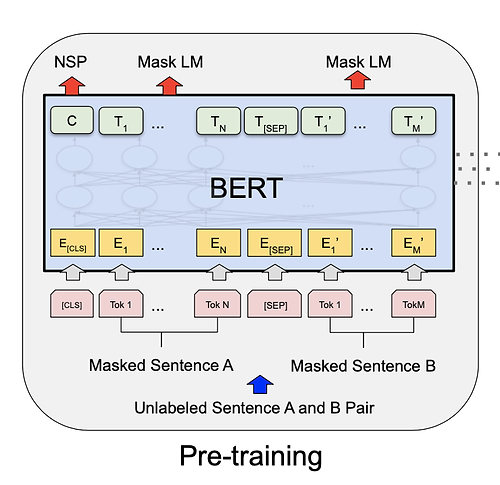
1. A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.
2. A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2.
3. A positional embedding is added to each token to indicate its position in the sequence. The concept and implementation of positional embedding are presented in the Transformer paper.



To predict if the second sentence is certainly connected to the first, the following steps are performed:

1. The entire input sequence goes through the Transformer model.
2. The output of the [CLS] token is transformed into a 2×1 shaped vector, using a simple classification layer (learned matrices of weights and biases).
3. Calculating the probability of IsNextSequence with softmax.

When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with the goal of minimizing the combined loss function of the two strategies. Summing both Masked LM and Next Sentence Prediction tasks, below is the architecture pre-training in BERT.

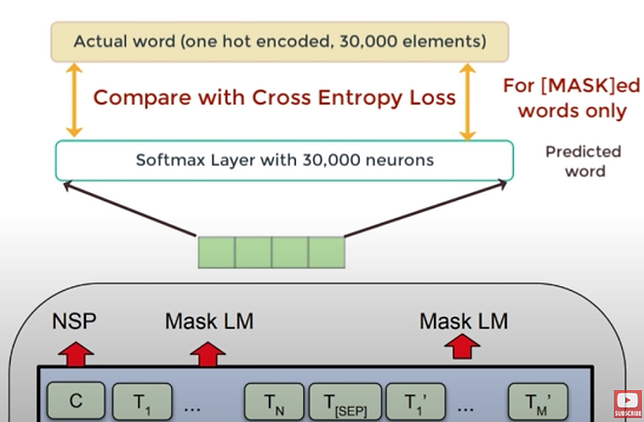


Pre-training architecture in BERT. [Credits](https://www.researchgate.net/figure/BERT-architecture-1_fig1_340295341)

**4.4 How training is done in BERT?**

Adding the three vectors gives embedding vector which is used as input to BERT. Segment and Positional embedding is required for temporal ordering since all these vectors are fed simultaneously to BERT so it's good for a language model to know the ordering of the words. The output is C(a binary value) and a bunch of other word vectors. For training, we need to minimize loss.

We need to take each Ti word vector, pass it to a fully connected layered output with the same number of neurons which is equal to the number of tokens in the vocabulary. Then apply the softmax activation function, this way we convert a word vector into a distribution. The actual word of the distribution would be a one-hot encoded vector for all the actual words. We then compare the two distributions and then train the network using cross-entropy loss.



Note that the output has all the words even though these inputs weren't masked at all. The loss function only considers the prediction of the masked words and ignores all the other words output by the network. This is done to ensure more focus is given in predicting these masked words, so it predicts them correctly and increases context awareness.

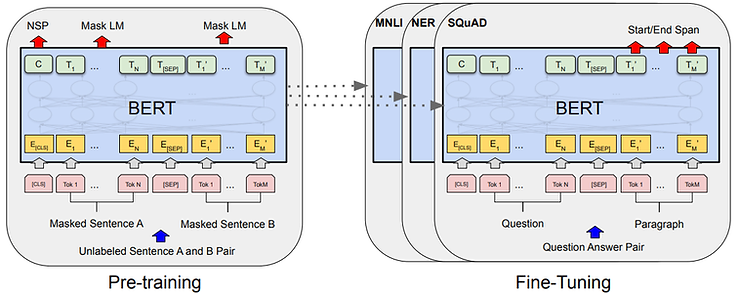
**4.5 How to use BERT (Fine-tuning)**

Fine-tuning is straightforward since the self-attention mechanism in the Transformer allows BERT to model many downstream tasks— whether they involve single text or text pairs—by swapping out the appropriate inputs and outputs.

For applications involving text pairs, a common pattern is to independently encode text pairs before applying bidirectional cross attention. BERT instead uses the self-attention mechanism to unify these two stages, as encoding a concatenated text pair with self-attention effectively includes bidirectional cross attention between two sentences.

BERT can be used for a wide variety of language tasks, while only adding a small layer to the core model:

1. Classification tasks such as sentiment analysis are done similarly to Next Sentence classification, by adding a classification layer on top of the Transformer output for the [CLS] token.
2. In Question Answering tasks (e.g. SQuAD v1.1), the software receives a question regarding a text sequence and is required to mark the answer in the sequence. Using BERT, a Q&A model can be trained by learning two extra vectors that mark the beginning and the end of the answer.
3. In Named Entity Recognition (NER), the software receives a text sequence and is required to mark the various types of entities (Person, Organization, Date, etc) that appear in the text. Using BERT, a NER model can be trained by feeding the output vector of each token into a classification layer that predicts the NER label.



Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different downstream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

**4.6 Definition of NLP:**

NLP deals with the ability of computers or machines to process, interpret, and generate human language. It encompasses tasks such as text analysis, sentiment analysis, language translation, and speech recognition. Role of NLP in Identifying Inappropriate Language and Hate Speech In the digital age, online communication has surged, but so has the prevalence of offensive content, including hate speech and cyberbullying.

**4.61 NLP plays a crucial role in addressing this issue:**

* Text Classification: NLP models can categorize text into different classes, such as normal language, offensive language, or hate speech.
* Feature Extraction: NLP extracts relevant features from text, enabling machine learning models to learn patterns associated with inappropriate content.
* Contextual Understanding: NLP models consider context, semantics, and syntax to understand the meaning behind words and phrases.
* Rephrasing and Moderation: When offensive content is detected.
* Removal: Offensive terms can be removed.
* Rephrasing: NLP can rephrase the content while preserving its original meaning.
* Machine Learning Models: NLP techniques are combined with machine learning algorithms to create effective hate speech detection systems.
* Multilingual Support: NLP models can handle multiple languages, making them adaptable to diverse online platforms.

**4.62 Challenges and Impact:**

* Arbitrary Definitions: Defining what is “inappropriate” varies across cultures and contexts.
* Anonymity: Online interactions often occur anonymously, leading to increased offensive language.
* Community Well-Being: Detecting and moderating hate speech is essential for maintaining respectful online communities and safeguarding users.

**4.63 Applications:**

* Social Media Platforms: NLP aids in identifying hate speech on platforms like Twitter, Facebook, and Instagram.
* Parental Filters: NLP-based filters can protect children from inappropriate content.
* Semantic Integrity: NLP ensures that rephrased content maintains its semantic integrity.

**4.7 Role of Anomaly Detection in Hate Speech**

Anomaly detection algorithms are instrumental in combating the presence of inappropriate language and hate speech in online platforms. By harnessing the power of AI, these algorithms analyze patterns and identify deviations from normal language usage, allowing for the detection and filtering of harmful content. This proactive approach contributes to the creation of safer and more inclusive digital spaces.

To begin, anomaly detection algorithms are trained on large datasets that encompass a wide range of language usage, including both normal and offensive content. By learning the patterns, structures, and context of normal language, these algorithms develop a baseline understanding of what constitutes acceptable communication. This understanding serves as a reference point for identifying anomalies or deviations that may indicate the presence of inappropriate language or hate speech.

When confronted with new text inputs, the anomaly detection algorithm compares the language patterns against the established baseline. If the algorithm detects a significant deviation from the expected patterns, it raises an alert, indicating the potential presence of harmful content. This alert prompts further review and action from the platform administrators, who can then evaluate the flagged content and take appropriate measures to address it, such as removing or moderating the offensive material.

By leveraging AI and anomaly detection algorithms, online platforms can proactively identify and address instances of inappropriate language and hate speech. This technology acts as an additional layer of defense, complementing human moderation efforts and contributing to the overall safety and inclusivity of online communities. Through continuous learning and refinement, these algorithms become more adept at recognizing nuanced forms of harmful content, helping to create a more positive and respectful online environment for all users.

In summary, anomaly detection algorithms utilize AI to analyze language patterns, detect deviations, and flag potentially inappropriate content. By working in tandem with human moderation efforts, these algorithms contribute to the creation of safer and more inclusive online spaces, fostering a positive and respectful digital community.

**4.5 DATA FLOW DIAGRAM:**

Creating a data flow diagram (DFD) for identifying inappropriate language and hate speech using AI involves illustrating how data moves through the system. Here's a basic outline for such a diagram:

**1.** **External Entities**: These represent external sources of data or users interacting with the system.

- Users (Inputting text data)

- AI Model (Processing the text data)

**2.** **Processes**: These represent the actions taken by the system.

- Text Preprocessing (Cleaning and formatting the text data)

- AI Algorithm (Applying machine learning algorithms to classify the text)

- Decision Making (Determining if the text contains inappropriate language or hate speech)

**3. Data Stores**: These represent where data is stored within the system.

- Input Data Store (Storing the raw text data input by users)

- Training Data Store (Storing the labelled training data used to train the AI model)

- Output Data Store (Storing the results of the AI's classification)

**4. Data Flows**: These represent the movement of data between the external entities, processes, and data stores.

- User Input to Text Preprocessing

- Text Preprocessing to AI Algorithm

- AI Algorithm to Decision Making

- Decision Making to Output Data Store

**5. Annotations**: Use labels to describe the data flows, processes, and data stores in the diagram, such as "Input Text Data," "Cleaned Text Data," "Classification Result," etc.

**TEXT PREPROCESSING**

**OUTPUT DATA STORE**

**DECISION MAKING**

.

**USER INPUT**

**TEXT PREPROCESSING**

**AI ALGORITHM**

**DECISION MAKING**

**OUTPUT DATA STORE**

In the block diagram, each block represents a component or process in the system, and the arrows indicate the flow of data between them. This diagram provides a visual representation of how data moves through the system for identifying inappropriate language and hate speech using AI.

**4.6 USE CASE DIAGRAM:**

**User Interaction**: Represents the user's interaction with the system.

**Identify Inappropriate Language and Hate Speech Text**: Use case where the system processes the text input to identify inappropriate language and hate speech.

**Provide Classification Result to User**: Use case where the system provides the classification result to the user.

This detailed use case diagram highlights the specific actions and interactions involved in the process of identifying inappropriate language and hate speech using AI, from user input to providing the classification result.

In the use case diagram, "User Interaction" represents the interaction between the user and the system for identifying inappropriate language and hate speech. The "Identify Inappropriate Language and Hate Speech" use case encompasses the entire process of text preprocessing, AI algorithm application, and decision-making.

This detailed use case diagram highlights the specific actions and interactions involved in the process of identifying inappropriate language and hate speech using AI, from user input to providing the classification result.

USER INTERACTION

PROVIDE CLASSIFICATION RESULTS TO USER

IDENTIFY INAPPRORIATE LANGUAGE AND HATE SPEECH TEXT

OUTPUT DATA STORED

DECISION MAKIG

AI ALGORITHM

USER INPUT

TEXT PRE PROCESSING

**CHAPTER-5**

**IMPLEMENTATION**

**5.1 Introduction to Python**

Python is a versatile, high-level, **interpreted programming language** created by **Guido van Rossum** in the late 1980s at the National Research Institute for Mathematics and Computer Science in the Netherlands. Let’s delve into the key aspects of Python

These are some facts about Python. Python is present most widely used multi-purpose and high-level programming language. It allows programming in Object-Oriented and Procedural paradigms. Python programs commonly are smaller than other programming languages like Java. Programmers have to write relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by the almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python language is huge collection of standard library it can be used for the following Machine Learning.

* GUI applications (such as Kivy, Tkinter, PyQt, etc.)
* Web frameworks such as Django (used by YouTube, Instagram, Dropbox)
* Image processing (such as Opencv and Pillow)
* Web scraping (such as Scrapy, BeautifulSoup and Selenium)
* Test Framework
* Multimedia

**5.2 History of Python**

* Python was named after the **comedy television show Monty Python’s Flying Circus**.
* It gained popularity due to its readability and expressiveness.
* Guido van Rossum released the first version (Python 0.9.0) in **1991**.
* The Python Software Foundation further developed it.

**5.3 Features of Python**

* **Easy to Learn and Use:**

Python’s syntax is straightforward, making it ideal for beginners. It emphasizes code readability, allowing programmers to express concepts concisely.

* **Interpreted Language:**

Python doesn’t require compilation; it’s interpreted line-by-line. This simplifies debugging and makes it suitable for newcomers.

* **Cross-Platform Compatibility:**

Python runs seamlessly on various platforms: **Windows, Linux, Unix, and Macintosh**. It’s a portable language.

* **Free and Open Source:**

The Python interpreter is open-source, making it free to install, use, and distribute.

* **Object-Oriented:**

Python supports object-oriented programming with classes and objects.

* **GUI Programming Support:**

You can create graphical user interfaces (GUIs) using Python.

* **Integration Capabilities:**

Easily integrate Python with other languages like **C, C++, and Java**.

**5.4 Python Applications**

Python’s versatility extends across various domains:

* **Web Applications:**

Python powers many web frameworks (e.g., **Django, Flask**). It’s widely used for web development.

* **Desktop GUI Applications:**

Create desktop applications with Python using libraries like **Tkinter**.

* **Software Development:**

Python is employed in software development across industries.

* **Scientific and Numeric Computing:**

Libraries like **NumPy, SciPy, and Pandas** facilitate data analysis and scientific computations.

* **Business Applications:**

Python is used for automating tasks, data processing, and reporting.

## ****5.5 Advantages of Python****

* **Easy to Read, Learn, and Code:**

Python’s **simple syntax** resembles English, making it beginner-friendly. No need for semicolons or braces. Maintenance costs are low due to its readability.

* **Dynamic Typing:**

Variables’ data types are assigned automatically during runtime. Facilitates dynamic coding.

* **Free and Open Source:**

Python is **free** and has an **open-source license**. Modify and distribute code without restrictions.

* **Platform-Independent (Write Once, Run Anywhere):**

Code written on one OS (e.g., Windows, Mac, Linux) runs on others without changes. Be cautious with system-dependent features.

* **Extensive Third-Party Libraries:**

Python boasts libraries like **NumPy, Pandas, Tkinter, and Django**. Simplifies coding with built-in functions and algorithms.

* 1. **Disadvantages of python**
* **Slow at Runtime**:

Python is an interpreted language, which makes it slower compared to compiled languages like C/C++ or Java.

* **Mobile Application Development:**

While Python excels in desktop and server platforms, it is not a strong choice for mobile development.

* **Difficulty in Using Other Languages:**

Python enthusiasts may find it challenging to switch to other programming languages.

* **High Memory Consumption:**

Python’s memory usage can be high due to its flexible data types. For memory-intensive tasks, Python may not be the best choice.

* **Not Widely Used in Enterprise Development:**

Python is robust and stress-free, but it lacks popularity in large enterprises.

* 1. **Getting Started with Python**

**5.71 Installation:**

* Install Python (preferably **Python 3**).
* Check out the Python Installation Tutorial.
  1. **Running Python Programs:**
* Use the interactive interpreter prompt or execute script files.
* Python 3 is recommended for its semantic correctness and modern features.
* Remember, Python’s simplicity and versatility make it an excellent choice for both beginners and experienced developers

.

**5.8 Install Python on Windows and Mac Step by Step:**

**Step 1:** Go to the official site and use Google Chrome or any other web browser to download and install Python. Or click on the following link:

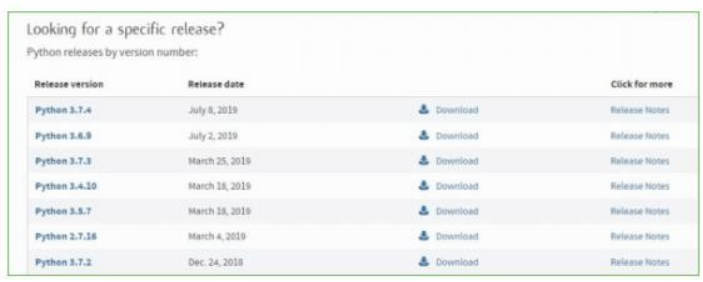
https://www.python.org

|  |
| --- |
|  |
|  |  |

**Step 2:** Click on the Download Tab.

|  |
| --- |
|  |
|  | C:\Users\Hxtreme\AppData\Local\Temp\ksohtml8132\wps2.png |

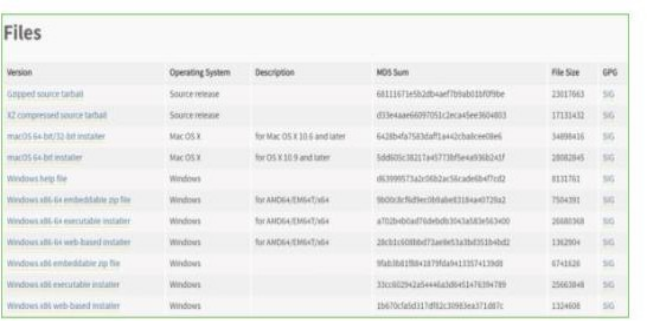
**Step 3**: You can select the yellow Download Python for Windows 3.7.4 button, or you can scroll down and click the download of the corresponding version. Here, we are downloading the latest version of Python for Window



|  |
| --- |
|  |
|  |  |

**Step 4:** Scroll down the page until you find the "File" option.

**Step 5:** Here you will see different versions of Python and operating systems.



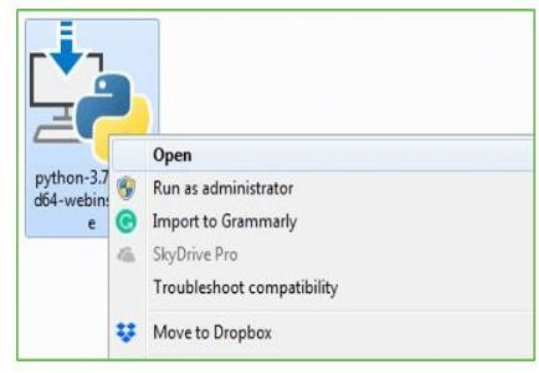
To download Windows 32bit Python, you can select the built-in Windows X86 Zip file, Windows X86 executable installer or Windows X86 installer on the Web.

To download Windows 64bit Python, you can select any option of the three options. Zip File Windows X866, Windows X8664 executable installer or Windows X8664 installer based on the web. Here you will install the installer based on the Windows X8664 website. Here, the first part of the Python version was completed. Now we will go in advance the second part when installing Python I. Note: You can click on the option of the release of the version to know the changes or updates made in the version.

**Installation of Python**

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process

**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7



|  |
| --- |
|  |
|  | C:\Users\Hxtreme\AppData\Local\Temp\ksohtml8132\wps6.png |

**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.

|  |
| --- |
|  |
|  | C:\Users\Hxtreme\AppData\Local\Temp\ksohtml8132\wps8.png |

**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python – V** and press Enter.

|  |
| --- |
|  |
|  | C:\Users\Hxtreme\AppData\Local\Temp\ksohtml8132\wps9.png |

**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

**Check how the Python IDLE works Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.

|  |
| --- |
|  |
|  | C:\Users\Hxtreme\AppData\Local\Temp\ksohtml8132\wps10.png |

**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Creating a Flask Backend (app.py)**

1. **Flask Web Application:**
   * Your app.py script serves as the backend for your hate speech detection system.
   * It’s where you’ll handle incoming requests and communicate with the Hugging Face API.
2. **Route Definition:**
   * Define a route at the root URL (“/”) that accepts both GET and POST requests.
   * When a POST request is received (i.e., the user submits text), follow these steps:
     + Retrieve the input text from the form.
     + Send the text to the Hugging Face API for inference using the query function.
     + Return the JSON output from the API.
   * When a GET request is received (i.e., the user accesses the page), render the index.html template.
3. **The query Function:**
   * Implement the query function:
     + It sends a POST request to the Hugging Face API with the provided payload (input text).
     + The API will perform hate speech detection and return a JSON response containing predictions.

**Creating an HTML Frontend (index.html)**

1. **Simple Web Page:**
   * Your index.html file creates a straightforward web page for hate speech detection.
   * Keep it minimal and user-friendly.
2. **Form and Textarea:**
   * Include a form with a textarea where users can input text.
   * When the form is submitted, JavaScript intercepts the submission event, prevents the default form behavior, and sends a POST request to the Flask backend.
3. **JavaScript Interaction:**
   * Upon receiving the JSON response from the backend, the JavaScript function displayOutput is called.
   * In this function:
     + Dynamically create HTML elements (e.g., divs) to display the output (labels and scores).
     + Populate these elements with the relevant information.

**CSS Styling**

1. **Visual Appearance:**
   * Define CSS styles to control the appearance of your web page.
   * Set font families, background colors, margins, paddings, and box shadows to create a visually appealing layout.
2. **Element-Specific Styling:**
   * Apply specific styling to individual elements.

**Setting Up a Virtual Environment and Running Your Python Program**

1. Open Terminal in the desired directory.
2. Navigate to your project directory by typing cd inappropriate and pressing Tab, then Enter.
3. Create a new virtual environment by executing python -m venv venv.
4. Activate the virtual environment by running venv\Scripts\activate.bat.
5. Once inside the virtual environment, install the necessary dependencies with pip install flask and pip install requests.
6. Run your Python program using python app.py.
7. Finally, click the provided URL (e.g., “Running on http://127.0.0.1:5000”) to access your application.

**Sample Code**

from flask import Flask, render\_template, request, jsonify

import requests

app = Flask(\_\_name\_\_)

API\_URL = "https://api-inference.huggingface.co/models/Hate-speech-CNERG/bert-base-uncased-hatexplain"

API\_KEY = "hf\_pRUXpUjpZIEwabNavEQqbwJLcPKIQEXeCO"

headers = {"Authorization": f"Bearer {API\_KEY}"}

def query(payload):

    response = requests.post(API\_URL, headers=headers, json=payload)

    return response.json()

@app.route("/", methods=["GET", "POST"])

def index():

    if request.method == "POST":

        text\_input = request.form.get("text\_input")

        output = query({"inputs": text\_input})

        return jsonify(output)

    return render\_template("index.html")

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

**index.html**

!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Hate Speech Detection</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f8f9fa; /\* Light gray background \*/

            margin: 0;

            padding: 0;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

            flex-direction: column;

        }

        h1 {

            color: #333; /\* Dark gray heading \*/

            margin-bottom: 20px;

        }

        #input\_form {

            background-color: #fff; /\* White background \*/

            padding: 20px;

            border-radius: 8px;

            box-shadow: 0px 0px 10px 0px rgba(0,0,0,0.1); /\* Soft shadow effect \*/

            margin-bottom: 20px;

        }

        label {

            display: block;

            margin-bottom: 10px;

        }

        textarea {

            width: 100%;

            padding: 10px;

            margin-bottom: 10px;

            box-sizing: border-box;

            border: 1px solid #ccc; /\* Light gray border \*/

            border-radius: 5px;

            resize: none; /\* Disable textarea resizing \*/

        }

        button {

            background-color: #007bff; /\* Blue button \*/

            color: #fff;

            padding: 10px 20px;

            border: none;

            border-radius: 5px;

            cursor: pointer;

            transition: background-color 0.3s; /\* Smooth transition \*/

        }

        button:hover {

            background-color: #0056b3; /\* Darker blue on hover \*/

        }

        #output {

            background-color: #fff; /\* White background \*/

            padding: 20px;

            border-radius: 8px;

            box-shadow: 0px 0px 10px 0px rgba(0,0,0,0.1); /\* Soft shadow effect \*/

            white-space: pre-wrap; /\* Preserve line breaks \*/

        }

        .output-item {

            margin-bottom: 10px;

            padding: 10px;

            border-radius: 5px;

            background-color: #f5f5f5; /\* Light gray background for output items \*/

        }

    </style>

</head>

<body>

    <h1>Hate Speech Detection</h1>

    <form id="input\_form" method="post" action="/">

        <label for="text\_input">Enter your text:</label>

        <textarea id="text\_input" name="text\_input" rows="4" cols="50"></textarea>

        <button type="submit">Submit</button>

    </form>

    <div id="output"></div>

    <script>

        document.getElementById("input\_form").addEventListener("submit", async function(event) {

            event.preventDefault();

            const formData = new FormData(this);

            const response = await fetch("/", {

                method: "POST",

                body: formData

            });

            const result = await response.json();

            displayOutput(result);

        });

        function displayOutput(result) {

            const outputDiv = document.getElementById("output");

            outputDiv.innerHTML = ""; // Clear previous content

            result[0].forEach(item => {

                const outputItem = document.createElement("div");

                outputItem.classList.add("output-item");

                outputItem.innerHTML = `<strong>${item.label}:</strong> ${item.score}`;

                outputDiv.appendChild(outputItem);

            });

        }

    </script>

</body>

</html>

**CHAPTER 6**

**TESTING**

Testing in the context of identifying inappropriate language and hate speech using AI refers to the process of evaluating the performance, accuracy, and effectiveness of AI models designed for this purpose. Testing plays a crucial role in ensuring that the AI models can accurately detect and classify inappropriate language and hate speech in text data. Here are some key aspects of testing in this context:

1. Data Preparation: Before testing can begin, it's essential to prepare a diverse and representative dataset that contains examples of inappropriate language and hate speech. This dataset is used to train the AI model and evaluate its performance.

2. Training and Validation: The AI model is trained using the prepared dataset, and its performance is evaluated using a separate validation dataset. This step helps ensure that the model generalizes well to new, unseen data.

3.Testing Strategies: Various testing strategies can be employed, including unit testing, integration testing, and end-to-end testing. These strategies help identify and fix issues at different stages of the AI model development process.

4. Evaluation Metrics: To measure the performance of the AI model, evaluation metrics such as precision, recall, and F1 score are commonly used. These metrics help quantify the model's accuracy in identifying inappropriate language and hate speech.

5. Cross-Validation: Cross-validation is a technique used to assess the performance of the AI model by splitting the dataset into multiple subsets. The model is trained and tested on each subset to ensure robustness and generalizability.

6. User Feedback: User feedback can be incorporated into the testing process to validate the model's performance. Users can provide input on whether the model correctly identifies inappropriate language and hate speech.

7. Continual Testing and Improvement: Testing is an ongoing process, and AI models should be continually tested and improved to ensure they remain effective in identifying inappropriate language and hate speech as language use evolves.

Overall, testing plays a critical role in the development and deployment of AI models for identifying inappropriate language and hate speech, helping to ensure their accuracy, reliability, and effectiveness in real-world applications.

**Testing in the context of identifying inappropriate language and hate speech using AI offers several benefits:**

1. Ensures Accuracy: Testing helps ensure that AI models accurately identify inappropriate language and hate speech, reducing false positives and false negatives.

2. Improves Model Performance: By identifying and fixing errors and weaknesses in the AI model, testing helps improve its overall performance and effectiveness.

3. Enhances User Trust: A well-tested AI model inspires confidence in users, increasing trust in its ability to accurately identify inappropriate language and hate speech.

4. Mitigates Bias: Testing can help identify and mitigate biases in the AI model, ensuring that it treats all language fairly and accurately.

5. Validates Model Robustness: Testing helps validate the robustness of the AI model by ensuring it performs well under different conditions and with varying types of input data.

6. Saves Time and Resources: Testing helps identify and fix issues early in the development process, saving time and resources that would otherwise be spent on correcting errors later on.

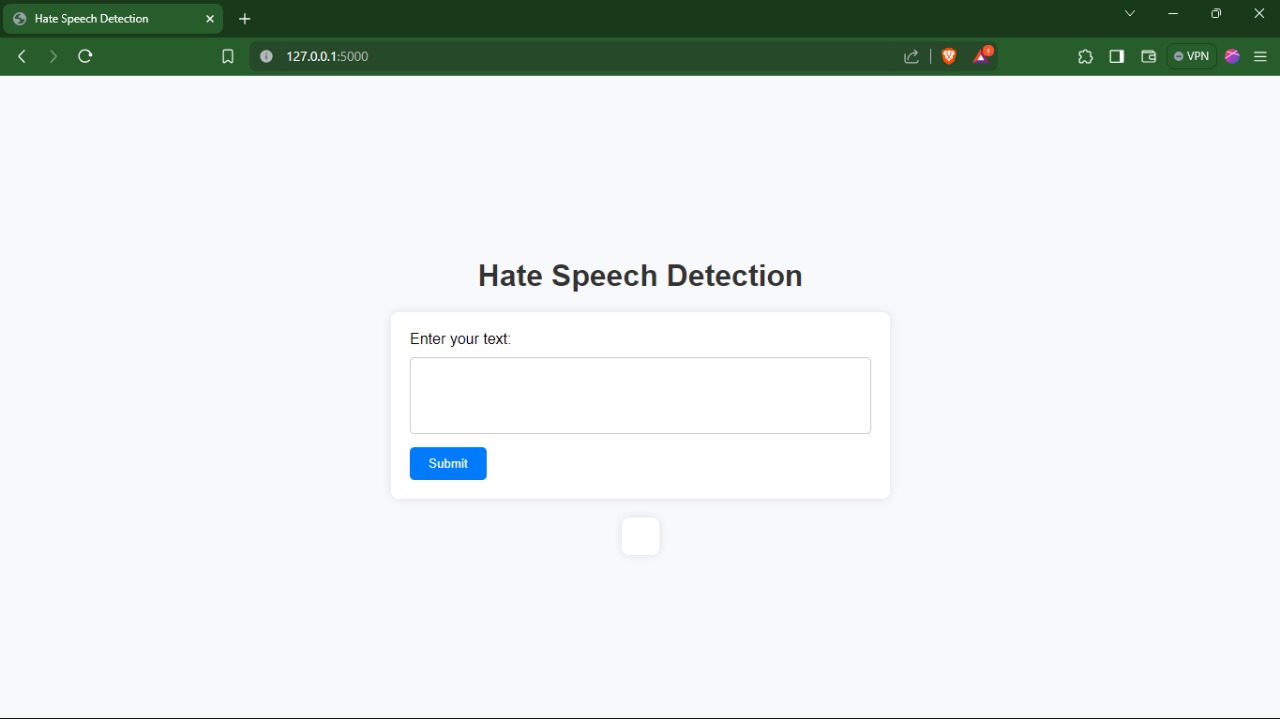
7. Compliance and Legal Requirements: Testing helps ensure that AI models comply with legal and regulatory requirements regarding the identification of inappropriate language and hate speech.

8. Continuous Improvement: Through iterative testing and feedback, AI models can be continuously improved to better identify inappropriate language and hate speech over time. Overall, testing is essential for ensuring the accuracy, reliability, and effectiveness of AI models in identifying inappropriate language and hate speech, leading to improved user trust and compliance with legal requirements.

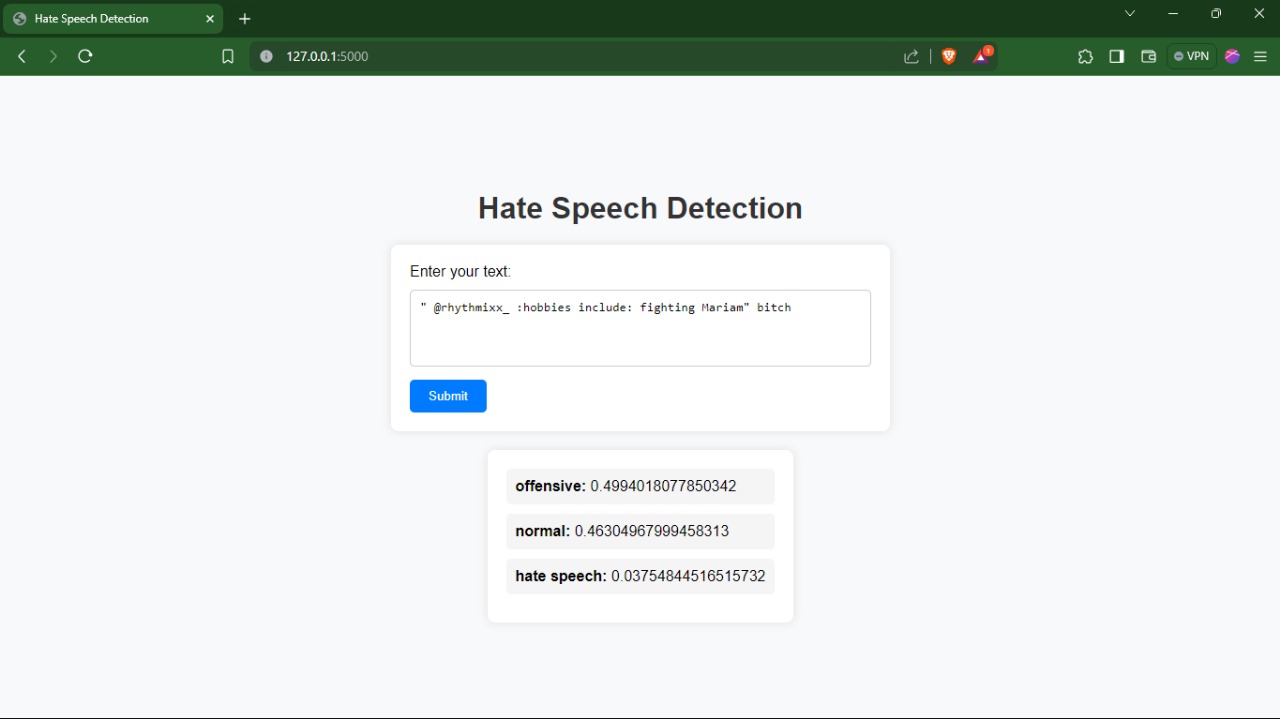
**CHAPTER 7**

**SAMPLE SCREENS**

**7.1 USER PAGE**

****

**7.2 RESULT**



**CHAPTER 8**

**CONCLUSION**

In conclusion, the use of AI in identifying inappropriate language and hate speech presents both significant opportunities and challenges. AI technologies, particularly those leveraging machine learning and natural language processing, offer powerful tools for automating the detection and moderation of harmful content online. These technologies can help platforms and organizations more effectively manage and enforce community guidelines, thereby fostering safer online environments.

However, the effectiveness of AI in this context is contingent upon several factors. These include the quality and diversity of the training data used to develop AI models, the sophistication of the algorithms employed, and the ability to adapt to evolving forms of inappropriate language and hate speech. Additionally, ethical considerations such as bias mitigation, transparency, and user privacy must be carefully addressed to ensure the responsible deployment of AI in content moderation.

While AI can significantly aid in identifying and moderating inappropriate content, user education and awareness are also crucial. Users should be informed about the importance of respectful online behaviour and the consequences of engaging in hate speech. Collaboration among tech companies, researchers, policymakers, and civil society organizations is essential for sharing best practices, developing common standards, and advancing AI technologies for content moderation.

Ethical considerations, such as transparency, accountability, and the protection of user privacy, should be at the forefront of AI development and deployment in content moderation. AI models for identifying inappropriate language and hate speech should be continually improved and adapted to address emerging trends and new forms of harmful content.

While AI can automate many aspects of content moderation, human review and intervention remain essential for nuanced decisions and addressing complex cases.AI models should be developed with a global perspective and cultural sensitivity to ensure that they are effective across different languages, regions, and cultural contexts.

Governments, industry stakeholders, and civil society should work together to establish responsible AI governance frameworks that promote the ethical and responsible use of AI in content moderation.

By addressing these aspects and working collaboratively, stakeholders can harness the potential of AI to effectively identify and mitigate inappropriate language and hate speech, fostering a safer and more inclusive online environment for all users.

Furthermore, the dynamic nature of language and the nuances of human communication present ongoing challenges for AI systems. Contextual understanding, sarcasm, and cultural references can all pose difficulties for AI in accurately identifying inappropriate language and hate speech. Continuous monitoring, evaluation, and refinement of AI models are therefore essential to maintain their effectiveness and relevance over time.

Despite these challenges, the potential benefits of AI in identifying inappropriate language and hate speech are substantial. By leveraging AI technologies responsibly and in conjunction with human oversight and intervention, platforms and organizations can enhance their content moderation efforts, promote healthier online discourse, and contribute to a safer and more inclusive online environment for all users.

**FUTURE ENHANCEMENT**

Future AI systems could focus on more fine-grained analysis of language, including identifying subtle forms of hate speech, implicit bias, and microaggressions. This would require more advanced NLP techniques to capture nuanced language patterns accurately.

Integrating emotion recognition capabilities into AI models can help detect hate speech that targets specific emotions, such as anger, fear, or sadness. This can provide a deeper understanding of the impact of hate speech on individuals.

Combining text analysis with other modalities such as images, videos, and audio can provide a more comprehensive understanding of content, especially in social media where hate speech can be communicated through various mediums.

AI systems can be enhanced to consider the context of the conversation or content, such as the platform, user history, and conversation dynamics, to make more informed decisions about whether content constitutes hate speech.

Future systems could be designed to interact with users to clarify intent or provide education on appropriate communication, aiming for proactive prevention of hate speech rather than just detection and removal.

AI models can be made more robust to adversarial attacks, where malicious actors attempt to evade detection by injecting subtle changes into their language. This involves developing techniques to detect and mitigate such attacks.

Enhancements could focus on developing AI models that can work across different platforms and languages, ensuring consistent and effective hate speech detection across diverse online environments.

Future enhancements should also address the ethical implications of AI in content moderation, ensuring transparency and accountability in decision-making processes and providing explanations for AI-driven decisions.

Developing metrics and methodologies to measure the impact and effectiveness of AI in reducing hate speech and its harmful effects on individuals and communities.

Establishing global collaboration frameworks and standards for hate speech detection using AI, facilitating knowledge sharing, and ensuring that AI systems are culturally sensitive and effective across different regions and languages.

To expand the topic of future enhancements in identifying inappropriate language and hate speech using AI for 9 pages, we can delve deeper into various aspects and technologies that could shape the future of this field. Here's an outline to guide the discussion:

Brief overview of the current state of AI in identifying inappropriate language and hate speech. Importance of continuous improvement and future enhancements in this area. Outline of the topics to be covered in the paper. Explanation of the challenges of context in language understanding.

Introduction to advanced NLP techniques like transformers for contextual understanding. Discussion on how future AI models can better capture nuances and context in language to improve accuracy. Importance of multilingual support in a global online environment.

Overview of current challenges in detecting hate speech and inappropriate language across multiple languages. Future enhancements in AI for better multilingual detection, including language-specific models and transfer learning approaches. Importance of real-time detection for timely content moderation. Discussion on enhancing AI models for faster processing and decision-making. Introduction to streaming NLP techniques and their application in real-time hate speech detection.

Explanation of bias in AI models and its impact on detecting hate speech. Overview of current approaches to bias mitigation in AI. Future strategies for reducing bias in AI models, including improved data collection, model architecture, and algorithmic fairness. Introduction to adaptive learning and its benefits in AI. Discussion on how adaptive learning can improve AI models for hate speech detection. Examples of adaptive learning techniques applied to NLP and AI.

Importance of user feedback in improving AI models. Challenges in integrating user feedback into AI systems. Future approaches to effectively incorporate user feedback for enhancing hate speech detection. Overview of privacy concerns in AI, particularly in hate speech detection. Introduction to privacy-preserving AI techniques. Future advancements in privacy protections for AI models, including federated learning and differential privacy.

Importance of collaboration in advancing AI for hate speech detection. Examples of successful collaborations and knowledge-sharing initiatives in the field. Future strategies for fostering collaboration and knowledge sharing among researchers, industry, and policymakers. Summary of key points discussed in the paper. Reflection on the potential impact of future enhancements in AI for identifying inappropriate language and hate speech. Call to action for continued research and development in this area.

This outline provides a comprehensive framework for exploring future enhancements in AI for identifying inappropriate language and hate speech, offering insights into the potential advancements and challenges in this evolving field.

These future enhancements aim to make AI systems more accurate, efficient, and ethical in identifying inappropriate language and hate speech, contributing to a safer and more inclusive online environment.

**BIBLOGRAPHY**

1. Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In Eleventh International AAAI Conference on Web and Social Media.
2. Fortuna, P., Nunes, S., & Rodrigues, F. (2018). A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 51(4), 1-30.
3. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016). Abusive language detection in online user content. In Proceedings of the 25th international conference on world wide web (pp. 145-153).
4. Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In Proceedings of the NAACL Student Research Workshop (pp. 88-93).
5. Park, J., & Fung, P. (2017). One-step and two-step classification for abusive language detection on Twitter. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 2091-2101).
6. Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media (pp. 1-10).
7. Basile, V., Caputo, A., Semeraro, G., & Rosso, P. (2019). Hate me, hate me not: Hate speech detection on Facebook. Information Processing & Management, 56(5), 1847-1860.
8. Huang, Y., Singh, V. K., & Atrey, P. K. (2019). Detecting hate speech in social media: A survey. ACM Computing Surveys (CSUR), 52(6), 1-38.
9. Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In Proceedings of the 26th International Conference on World Wide Web (pp. 1391-1399).
   1. .Malmasi, S., & Zampieri, M. (2018). Challenges in discriminating profanity from hate speech. In Proceedings of the Second Workshop on Abusive Language Online (pp. 3-11).