Lecture 1 - Introduction to IUI (1)

Ubiquitous Computing

- **Definition:** Ubiquitous computing refers to technologies that become so well integrated into daily life that they are virtually invisible to users.

> **Ubiquitous Computing Quote – Mark Weiser (1991)**
>
> "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."
> "…Hundreds of computers in a room could seem intimidating at first […] these hundreds of computers will come to be invisible to common awareness. People will simply use them unconsciously to accomplish everyday tasks."

The best technologies are those that operate in the background. Users interact with the services without needing to be aware of the technical details behind them.

> **Context on Ubiquitous Computing**
>
> Ubiquitous computing envisions a world where computing devices are embedded everywhere — in homes, workplaces, and public spaces — thus enabling seamless interactions without overwhelming the user with technical complexity.

Goals of Human-Machine Systems

Design Objectives:

- **Primary Goal:**
  - Design systems in which human-machine cooperation leads to performance that outperforms both humans and machines acting independently.
- **Key Challenge:**
  - Achieving intuitive cooperation between humans and computers.
- **Human-Human Communication as a Model:**
  - Techniques such as mid-air pointing and using natural gestures are highlighted as inspirations for current interface design.

Properties of Interactive Human-Centered AI

> **Definition of iHCAI**

An interactive human-centered artificial intelligence is an artificial intelligence that enables interactive exploration and manipulation in real time and is designed with a clear purpose for human benefit, while being transparent about who has control over data and algorithms.

Core Characteristics:

1. **Real-Time Interaction:**: Users can interact in real time with algorithms, models, and data, with immediate control over parameters.
2. **Instant Feedback:** Any changes made by the user can be observed immediately.
3. **Adjustable Processing Speed:** Systems may slow down fast processes to allow for user intervention.
4. **Interactive Exploration:** Users can explore the decision-making process, understanding how adjustments affect outcomes.
5. **Human Benefit:** The design should clearly state how humans benefit from the AI.
6. **Risk Awareness:** It should explain potential risks posed by the AI to both individuals and society.
7. **Control Transparency:** It must be clear who controls the AI—specifically, who has power over data, models, and algorithms.
8. **Data Transparency:** The sources of data, knowledge bases, and information used to inform the AI should be visible.

How can interfaces be designed to maximize the complementary strengths of humans and AI?

- **Human-AI Collaboration:** Interfaces should leverage AI's efficiency for data processing while allowing human oversight in decision-making.
- **Adaptive Interaction:** Implementing context-aware and explainable AI models ensures users can understand, guide, and correct AI-driven processes.

What ethical and societal challenges arise from increasingly autonomous AI systems?

- **Bias & Fairness:** AI decisions may reinforce biases present in training data, impacting fairness in areas like hiring, law enforcement, and healthcare.
- **Loss of Human Control:** Highly autonomous AI systems may reduce human oversight, raising concerns about accountability, decision justification, and unintended consequences.

How can transparency and accountability be ensured in complex, interactive AI systems?

- **Explainability:** AI models should provide human-readable justifications for decisions, allowing users to understand and challenge outcomes.
- **Regulatory Oversight:** Clear policies, audits, and human-in-the-loop frameworks are necessary to maintain control over AI-driven decision-making.

Designing principles for intervention user interfaces

- Ensure expectability and predictability
- Communicate options for intervention

- Allow easy exploration of interventions
- Easy reversal of automated intervention actions
- Minimize required attention
- Communicate how control is shared.

Lecture 2 - Introduction to IUI (2)

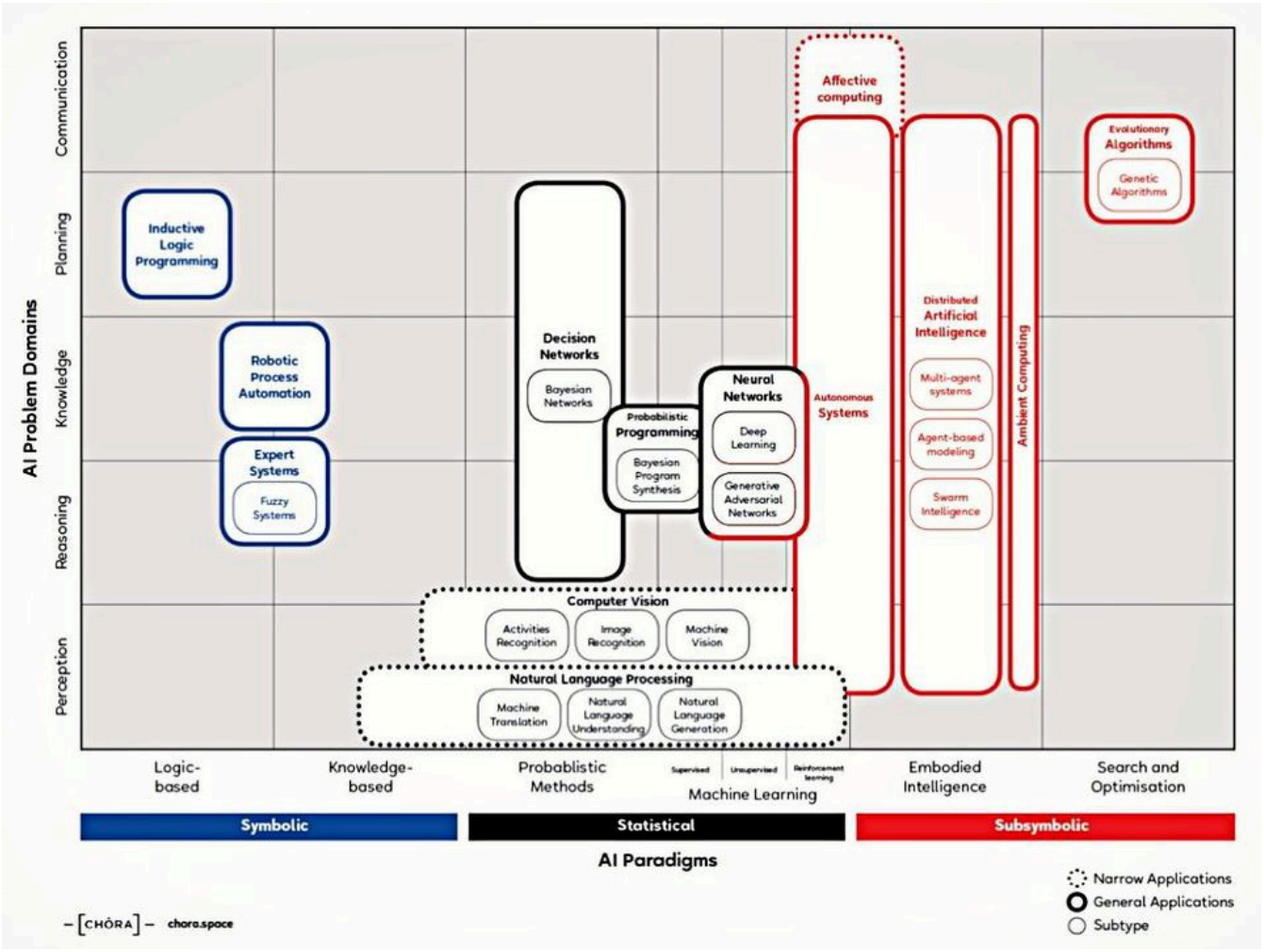
Defining Artificial Intelligence

- “What is considered Artificial Intelligence?” and “What is AI?”
  - **Artificial Narrow Intelligence (Weak AI):** Solves very specific, well-defined problems in a particular domain.
    - **Artificial General Intelligence (Strong AI):** Capable of mimicking human intelligence such that its behavior is indistinguishable from that of a human.
    - **Artificial Super Intelligence:** Surpasses human intelligence.

> **Understanding AI Categories**
>
> Distinguishing among narrow, general, and super AI is essential. Current real-world systems largely fall under narrow AI, which are specialized tools rather than all-encompassing thinking machines.

The AI Knowledge Map: Technologies and Domains

- **Technologies**
  - **Logic-based:** Tools for knowledge representation and problem-solving.
  - **Knowledge-based:** Systems using ontologies, databases, and rule sets.
  - **Probabilistic methods:** Approaches that handle uncertainty and incomplete information.
  - **Machine Learning:** Algorithms that learn patterns from data.
  - **Embodied Intelligence:** Systems that incorporate physical or simulated bodies to achieve higher-level intelligence.
  - **Search and Optimization:** Methods to intelligently explore and choose among many possible solutions.
- **Domains**
  - **Reasoning:** Solving problems through logical inference.
  - **Knowledge:** Representing and understanding the world.
  - **Planning:** Setting goals and achieving them.
  - **Communication:** Understanding and generating language.
  - **Perception:** Converting raw sensor data (e.g., images, sounds) into actionable information.

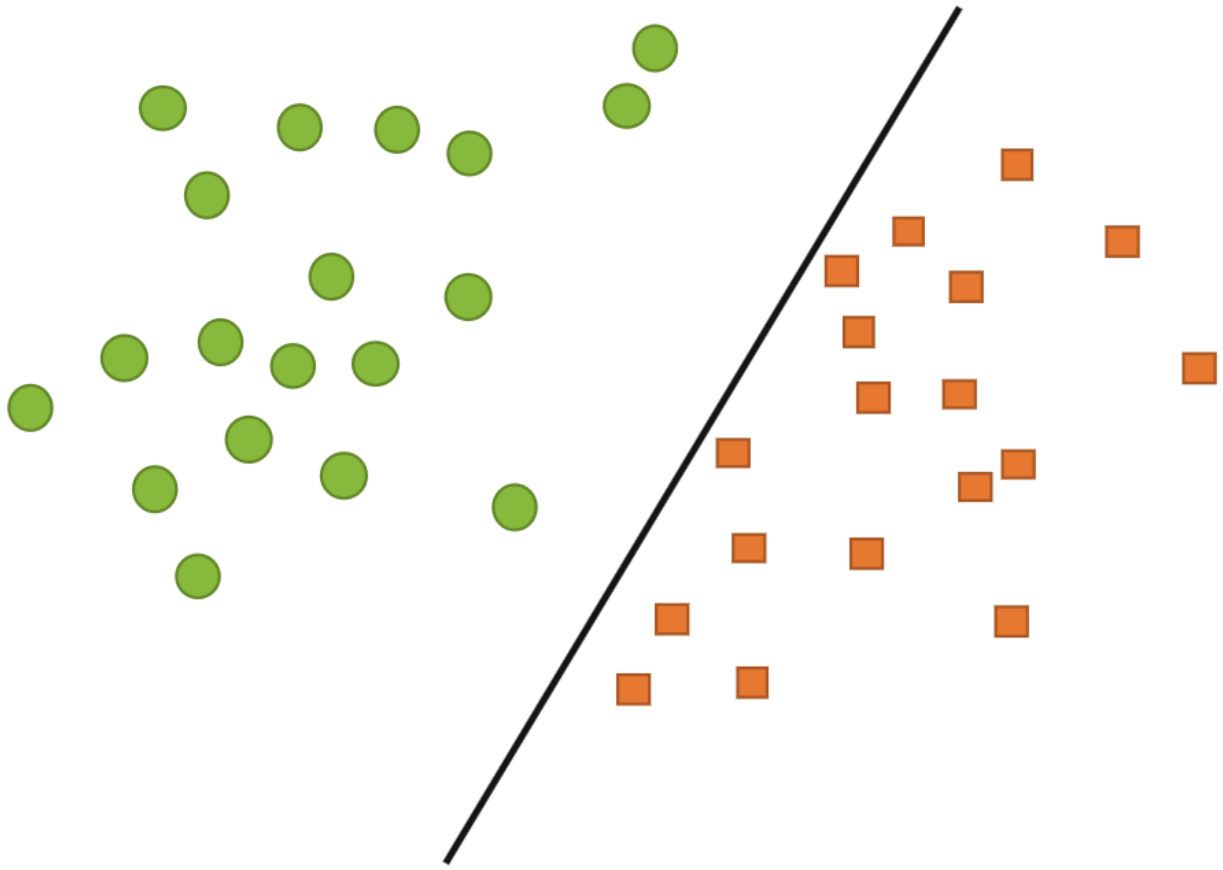


AI Knowledge Map: how to classify AI technologies. A sketch of a new AI technology landscape Francesco Corea

---

Classification in AI

- **Classification** as a fundamental problem in AI: determining a “line” that separates data into different groups (e.g., deciding whether an email is spam or not).
- **Key Points:**
  - Classification requires examples of data where the class (label) is known.
  - The goal is to find a boundary that maximally separates different classes.
  - **Example:** Support Vector Machines (SVM) are mentioned as an approach that finds the optimal boundary between classes.



How can we find a “line” that separates the two groups?

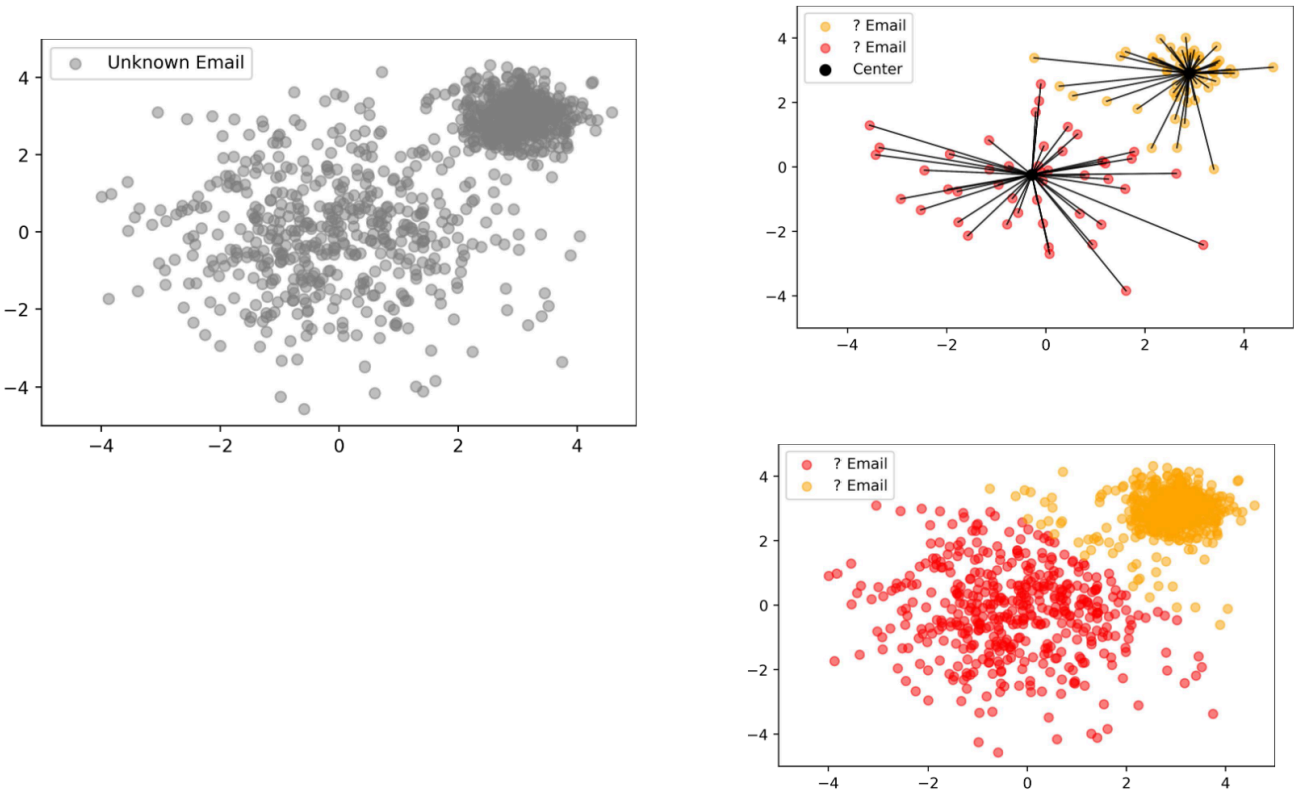
A common approach is to use a linear classifier such as a Support Vector Machine (SVM). The SVM finds the optimal hyperplane—essentially a line in two dimensions—that maximizes the margin between the two classes, thereby minimizing misclassification and providing a robust decision boundary.

Why Classification Matters

Classification is central to many AI applications — from spam detection to image recognition. It lays the groundwork for understanding supervised learning methods.

Supervised vs. Unsupervised Learning

- **Supervised Learning:**
  - Learning from labeled examples.
  - A dataset with known outcomes (labels) is used to train a model.
  - The model is then tested on unseen data to evaluate its performance.
- **Unsupervised Learning:**
  - Learning patterns from data without explicit labels.
  - Clustering, representation learning, and density estimation.
  - Used for exploratory data analysis and dimensionality reduction.



Unknown Data → K-Means Clustering → two clusters with each one center, every newly added points copuld shift the clusters

> **Unsupervised Learning**
>
> “The most common tasks within unsupervised learning are clustering, representation learning, and density estimation...”

> **Supervised vs. Unsupervised**
>
> Supervised learning relies on labeled data to predict outcomes, while unsupervised learning aims to uncover hidden structures in unlabeled data.

Unsupervised Learning and Clustering

	Supervised Learning	Unsupervised Learning
Discrete	Classification or Categorization	Clustering
Continuous	Regression	Dimensionality reduction

- **Learning Strategies:**

    - **Supervised:** Discrete outcomes (classification, regression).
        - **Unsupervised:** Discovering natural groupings (clustering) or reducing dimensionality.
- **Unsupervised Methods:**
    - Techniques such as hierarchical clustering, k-means clustering, Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and Independent Component Analysis
- **Clustering Focus:**
    - Detailed discussion on clustering as a method to uncover structure in data.
    - **K-means clustering** is highlighted with several slides discussing its application and challenges—such as determining the optimal number of clusters

---

Discussion: Usable Security

- **Security Applications:**
    - How to detect when “the wrong user” attempts to log in.
    - Differentiating between a denial-of-service attack and a user’s mistake (e.g., forgetting a password).
    - The potential use of clustering algorithms to identify anomalous behavior.

> **Usable Security Challenges**
>
> Machine learning can enhance security by automating anomaly detection, but careful tuning is required to avoid false positives and ensure user convenience.

Lecture 3 - Users' Context in Smart Environments

Ubiquitous Computing and the Vision for Disappearing Technology

Tactile Context:

The number of objects a person interacts with can inform designers about potential input channels and user engagement in a smart environment.

Ubiquitous Computing Quote – Mark Weiser

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."

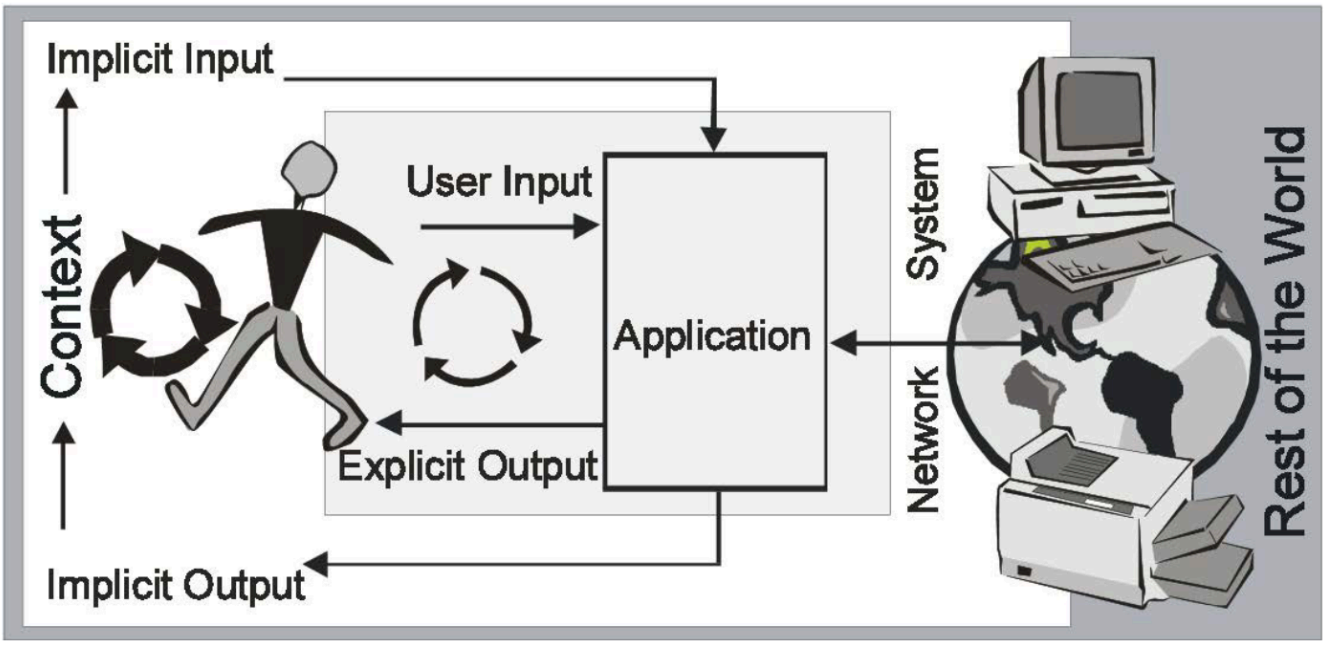
Computers embedded in everyday devices (light switches, thermostats, ovens) illustrate how technology becomes invisible to users while still enabling smart functionality.

Further Explanation:

Concept Explained by AI: Ubiquitous computing envisions a network of seamlessly integrated devices that support everyday tasks without drawing attention to themselves, thereby enhancing user experience by reducing cognitive load.

Design Considerations for Context-Aware Systems

- **Embedding Information and Interaction:**
  - Information is delivered at decision points (e.g., “What should I wear?”) without requiring explicit user action.
    - Over-provision of displays so that contextually relevant information is available.
    - Information must be unobtrusive and provided without forcing interaction.



Here **implicit inputs** (e.g., location, time, or activity) inform the application without deliberate user actions, while **explicit inputs** (e.g., voice commands or gestures) trigger direct responses.

> **Contextual Cues lead the designin process**
>
> This approach ensures that users benefit from contextual cues without being overwhelmed, leading to more natural and informed interactions.

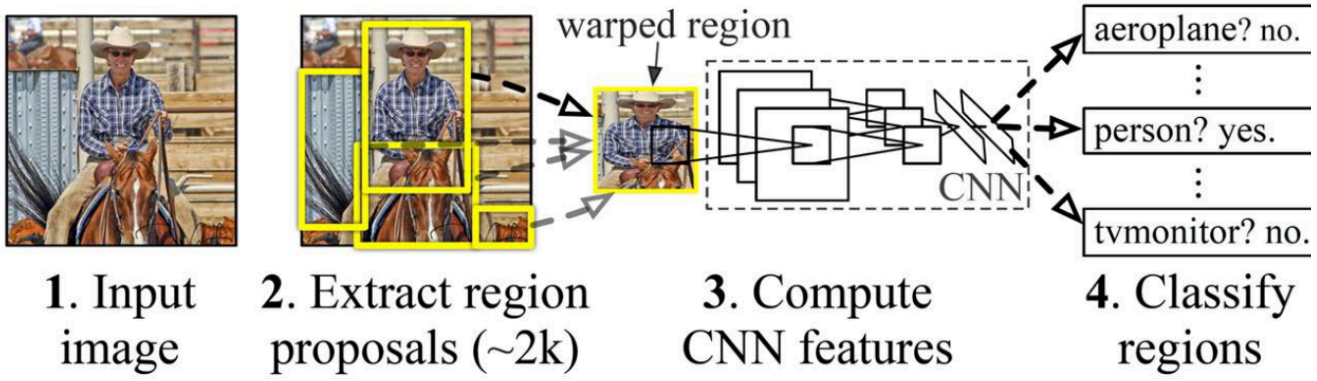
- **Output Adaptation:**
  - Use context (e.g., location, time, user activity) to adjust media quality, modality, content presentation, and notification timing.
- **Input Optimization:**
  - Simplify input through context-dependent defaults and context-sensitive menus (e.g., handwriting or speech recognition tailored to the current environment).

Physical Interaction and Context Sensing

Rethink Output	Rethink Input
Make use of context	Easing input by using context knowledge
Adjust media quality	Automate input (e.g., current time, meeting participants, document tracking)
Adapt media usage	Provide context-dependent defaults
Choose the modality	Optimize input space to fit the current context
Adapt content and visual representation	Use recognizers for handwriting/speech
Timing of output / notification	Implement context-sensitive menus
Interrupt at “appropriate” times	

Object Detection and Machine Learning for Context Extraction

Systems increasingly rely on **context extraction** to enhance user experiences. By leveraging **object detection and machine learning**, applications can interpret and react to real-world environments in real time. Context-aware computing aims to **reduce user effort** by making interfaces more adaptive, predictive, and intelligent.



Extracting Contextual Information

Context-aware systems gather data from multiple sources to better understand user needs. Key **contextual inputs** include:

- **GPS Location:** Determines where the user is, informing navigation or location-based services.
- **Voice Direction & User Activity:** Helps distinguish whether the user is speaking to a device or another person.
- **Emotion & Pose Recognition:** Uses facial expressions and body posture to infer user intent or state.
- **Surrounding Objects & Device Status:** Identifies real-world objects and device conditions (e.g., battery level) to adjust system behavior accordingly.

> **Context-Aware Systems Reduce Cognitive Load**
>
> By leveraging implicit input from the environment, systems can anticipate user needs, reducing the need for manual adjustments.

Object Detection Techniques

Object detection is fundamental in extracting context, allowing systems to recognize and classify visual elements in real-time. Two widely used techniques are:

R-CNN (Region-Based Convolutional Neural Networks)

- **How it Works:** Identifies regions of interest in an image and classifies them into objects.
- **Strengths:** High accuracy in object recognition.
- **Weaknesses:** Computationally expensive, slower inference time.

YOLO (You Only Look Once)

- **How it Works:** Uses a single neural network to process the entire image at once, making detections in real-time.
- **Strengths:** Fast, efficient, ideal for real-time applications (e.g., autonomous vehicles, AR systems).
- **Weaknesses:** Slightly less accurate than R-CNN in complex scenes with overlapping objects.

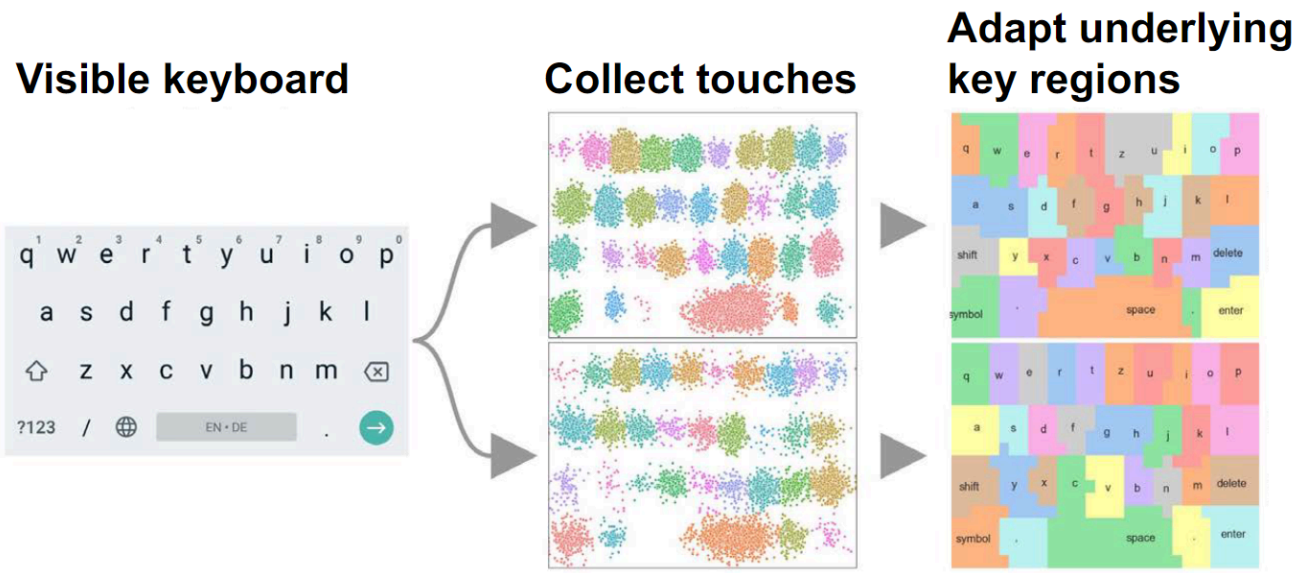
Method	Strengths	Weaknesses
R-CNN	High accuracy, robust detection	Slow inference, computationally expensive
YOLO	Fast, real-time detection, efficient	Slightly lower accuracy in complex scenes

---

Context-Aware Keyboards

Intelligent keyboard interfaces adapt based on user grip and hand position, enhancing usability. Key advancements include:

- **Palm Detection:** Adjusts keyboard layout if a user's palm is resting on the screen.
- **Finger Identification:** Recognizes which fingers are in use to optimize key spacing.
- **Grip Detection:** Identifies how a device is held, modifying key positions for better reachability.



These adaptations make touchscreens and virtual keyboards more responsive, reducing typing errors and improving accessibility.

> **Adaptive UI Design**
>
> Dynamic interfaces, like context-aware keyboards, demonstrate how **real-time sensor input** can optimize usability without requiring explicit user customization.

Machine Learning for Enhancing Sensing

To effectively interpret user behavior and surroundings, **machine learning models** are employed to process sensor data and make intelligent predictions. Common algorithms include:

Algorithm	Application Areas	Characteristics
Support Vector Machines (SVM)	Gesture recognition, speech processing	Works well with structured data
k-Nearest Neighbors (kNN)	Object detection, classification tasks	Simple, effective for small datasets
Decision Trees	Context classification, predictive models	Interpretable, but prone to overfitting
Random Forests	Activity recognition, image processing	More robust than individual trees
Gaussian Processes	Sensor fusion, uncertainty modeling	Handles uncertainty well

By applying these models, systems can **learn from user behavior**, allowing for **proactive** rather than **reactive** interaction design.

How can machine learning models be optimized for real-time context detection without compromising accuracy?

- **Model Optimization Techniques:** Use lightweight architectures (e.g., MobileNet, Tiny-YOLO) and techniques like quantization and pruning to reduce computational load while maintaining accuracy. **Efficient Data Processing:** Implement edge computing to process data locally, reducing latency and reliance on cloud services while using adaptive sampling to prioritize relevant sensor inputs dynamically.

Neural Network ## Neural Network Foundations for Context Sensing

Neural Network Basics

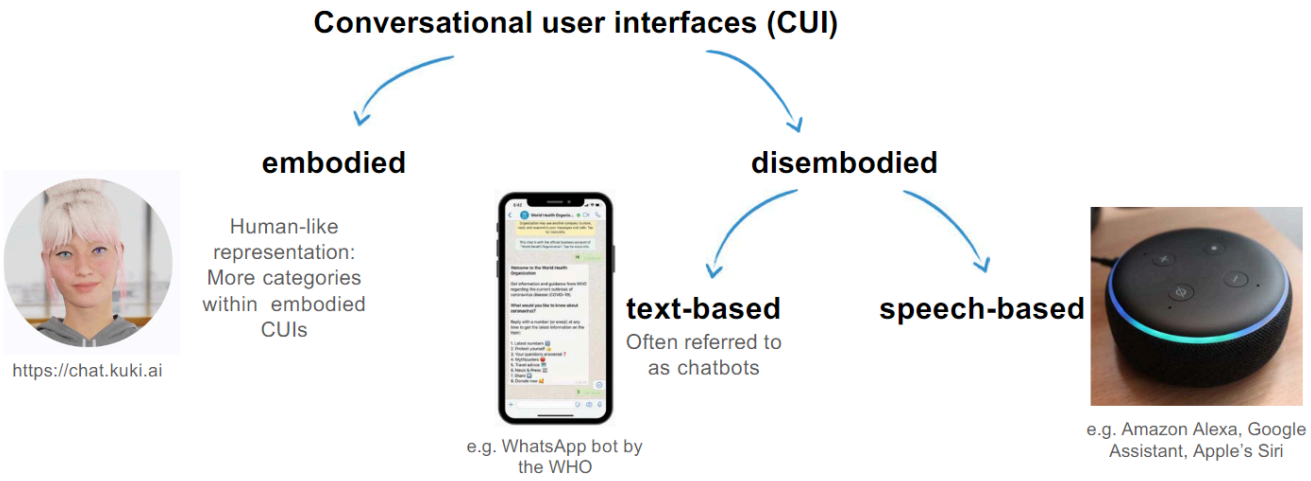
- **Structure:** Neural networks consist of perceptrons, dense layers, and pooling operations to process data efficiently.
- **Key Tasks:**
  - **Classification vs. Regression:** Predicts discrete labels (classification) or continuous values (regression).
  - **Feature Extraction vs. Representation Learning:** Compares manual feature engineering with networks learning patterns directly from data.
- **Common Architectures:** CNNs (image tasks), RNNs (sequential data), LSTMs (long-term dependencies).

Lecture 4 - Voice User Interfaces (VUI)

What are Voice User Interfaces?

A **Voice User Interface (VUI)** enables users to interact with systems using spoken language. Unlike **Graphical User Interfaces (GUIs)**, VUIs rely entirely on **speech recognition and voice synthesis**.

> **Definition of VUI - Cohen et al. (2004)**
>
> "A Voice User Interface (VUI) is what a person interacts with when communicating with a spoken language application."



Key Characteristics:

- Spoken **natural language interaction**.
- Hands-free and eyes-free usability.
- Often integrated with AI-driven **conversational agents**.

> **VUI**
>
> Language is the most crucial channel of communication between people

Key historical Milestones:

- **ELIZA (1964)** – Early chatbot simulating conversation.
- **IBM Watson (2006)** – Advanced AI-driven speech recognition.
- **Interactive Voice Response (IVR) Systems (~2000s)** – Automated phone-based speech systems.
- **Smart Assistants (2010s–present)** – Siri, Alexa, Google Assistant.

Advantages of VUIs:

- **Easy to learn** – Natural speech-based interaction.
- **Hands-free operation** – Useful in driving, cooking, accessibility.
- **Fast communication** – Speaking is often quicker than typing.

- **Accessibility**

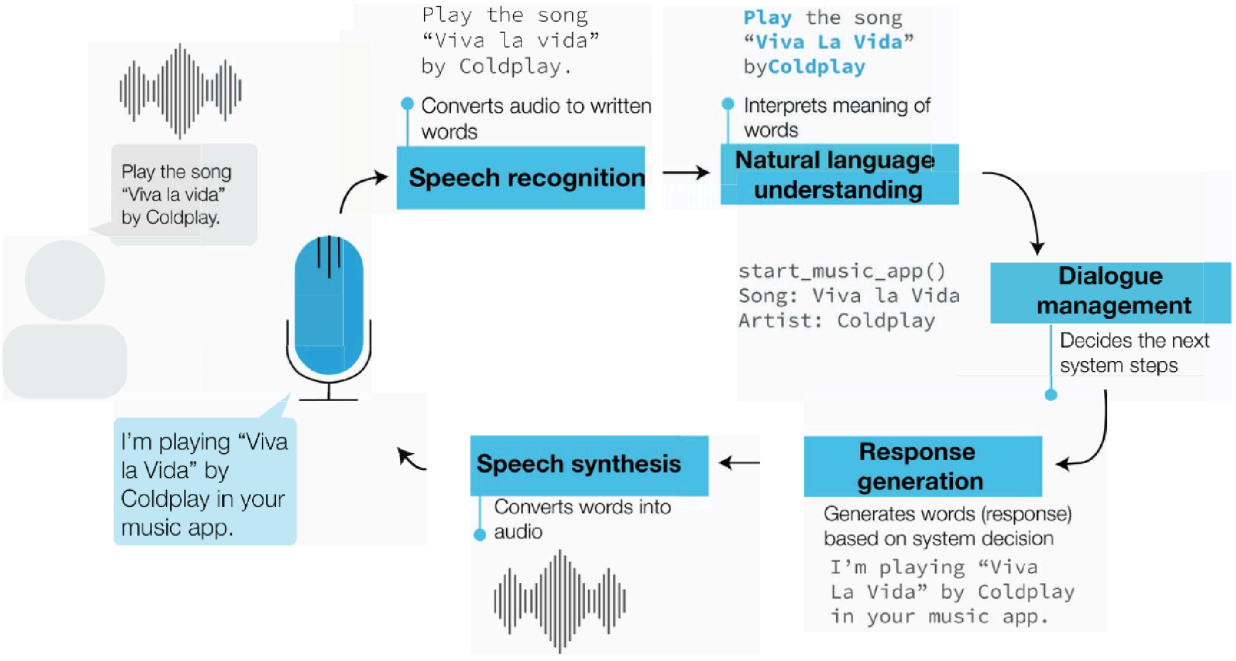
> **Speed of Speech vs. Typing**
>
> Speaking in English and Mandarin is nearly **3 times faster** than typing. *(Ruan et al. 2018)*

Challenges:

- **Speech recognition issues** – Accents, dialects, background noise.
  - **Homophones**: similar sounding words that have different meanings.
    - There vs their vs they're
    - Flower vs flour
  - Accents/dialects: Indian or Singapore English dialect
  - Speech disabilities: e.g., stuttering, Dysarthria (weakened speech muscles)
- **Lack of contextual understanding** – Difficulty with ambiguous phrases.
- **Privacy concerns** – Constant microphone listening raises security risks.
- Public and Group use can be difficult

> **Pragmatics**
>
> Language always needs to be understood in the context, like:
>
> - That's a typical example of a job well done
> - You've a green light

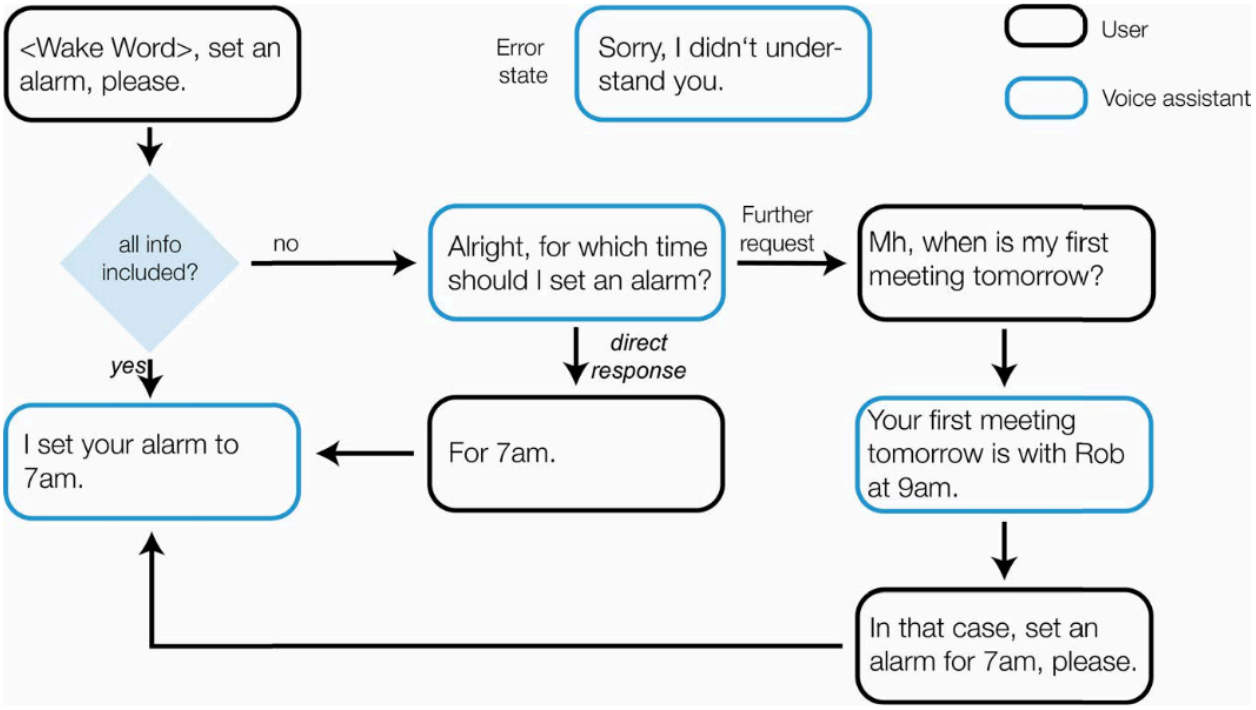
Design Principles for VUIs *(Slide 30-35)*



Best Practices:

1. **Guide users** – Indicate available commands clearly.

2. **Limit information overload** – Short responses improve usability.
   - inform users about their options
   - inform users about the current function; eg.: vui should not answer "done" but rather: "the timer has been set to 5min"
   - only list up to four items when giving feedback
3. **Confirm user input** – Reduce errors by repeating key responses.
4. **Use conversational flow** – Avoid robotic or rigid phrasing.
5. **Manage user expectations** – Clearly define what the VUI can and cannot do.



Short-Term Memory Limitations

Users can retain only **4-7 chunks** of spoken information. *(Miller, 1956)*

Humanizing VUI

- **Automatic Personality Attribution:**
  We naturally assign personalities to voice user interfaces (VUIs), even if none was intentionally designed. This subconscious process **affects user trust, likability, self-disclosure, and even purchasing decisions.**
- **Impact on User Behavior:**
  The personality perceived in a VUI can make users more inclined to interact, share personal details, or follow recommendations. Since many VUIs tend to adopt female voices, reinforcing stereotypes.

Gender Stereotypes in VUI Design

Historically, female voices have been perceived as more nurturing, approachable, and submissive, which may influence how users interact with the VUI. For example, users might be more willing to disclose personal information or follow suggestions from a female voice, simply because of these socially ingrained associations. While these design choices can enhance user comfort and engagement, they also

perpetuate gender norms and diminish the diversity of roles that both males and females can embody in automated systems. für einige wenige Sekunden

- **Avatar and Visual Feedback Options:**
  While using avatars can enhance emotional expression, they risk falling into the uncanny valley, where the human-like appearance becomes unsettling. Alternatively, non-anthropomorphic feedback (e.g., a glowing blue ring like the Alexa Dot) can communicate status without mimicking human traits.
- **Design Complexity and Ethical Considerations:**
  Designers face a dilemma: making a VUI as human as possible may boost engagement, but it also risks manipulation and reinforces gender stereotypes. The challenge lies in balancing intuitive, empathetic design with ethical transparency.

Should we make VUIs as human as possible?

- **Pro:** A more human-like VUI can foster natural interaction, build trust, and increase user satisfaction by mimicking real human conversation.
- **Contra:** Over-humanization might create unrealistic expectations, trigger discomfort (uncanny valley), and perpetuate stereotypes, ultimately compromising ethical design and user autonomy.

Challenges and Future Trends

Current Limitations:

- **Context interpretation** – Understanding **pragmatics** and **syntactic ambiguity**.
- **Emotional intelligence** – Recognizing tone and emotions remains difficult.
- **Bias in AI** – Some VUIs struggle with gender and language biases.

Future Directions:

- **Multimodal interaction** – Combining voice with gestures or visuals.
- **More personalized assistants** – Learning user preferences dynamically.
- **Ethical AI considerations** – Improving privacy and reducing biases.

Social Perception of VUIs - Reeves & Nass (1996)

"Users apply the same social rules to VUIs as they do to human conversations."

Lecture 5 - Introduction to NLP and Text Processing

Structured Summary

Natural Language Processing

- Natural Language Processing (NLP) combines computational techniques with linguistic theory to analyze and process human language.
- **Daily Use Cases**: Search engines (e.g., Google queries about U.S. presidents), transcription tools (e.g., Samsung Galaxy Note's handwriting-to-text).
- **Typical Tasks**:
  - Question answering (e.g., "Who is the 48th U.S. president?").
  - Entity recognition (e.g., identifying names like "Donald Trump").
  - Text normalization and translation.

Definitions and Concepts

- **Natural vs. Artificial Languages**: Natural languages evolve organically (e.g., English), while artificial languages follow strict rules (e.g., programming languages).
- **Semantics**: Meaning derived from words and sentences.
- **Pragmatics**: Contextual interpretation of language.

> **Definition of NLP, Liddy, E.D. (2001)**
>
> "Natural Language Processing is a theoretically motivated range of **computational techniques** for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of **achieving human-like language processing** for a **range of tasks or applications**."

Challenges for NLP:

- Paraphrasing, translation, question answering, and inference require understanding context and ambiguity.
- Example: Translating idioms (e.g., "break a leg") $\rightarrow$ cultural nuances.

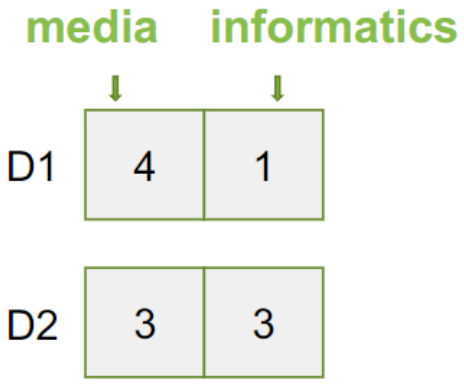
Algorithms and Techniques

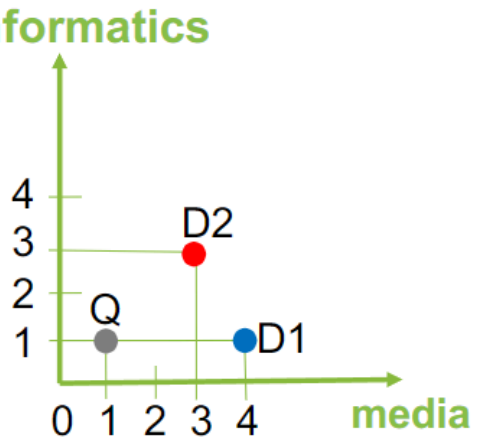
- **High-Level Tasks**: Sentiment analysis, summarization, topic modeling.
- **Low-Level Tasks**: Tokenization, stemming, stop word removal.

Python Example: Counting word occurrences

Converting text documents into word frequency vectors for search relevance.

- **Relevance Calculation:** By turning words into numerical features (counts), search algorithms can compare documents (e.g.: the distance between D1 and D2) to the query in a structured way.
- **Vector Space Model Foundation:** This is the basis of many Information Retrieval systems and a stepping stone toward more advanced **Natural Language Processing** techniques





Tokenization and Normalization

- **Tokenization**: Splitting text into words/tokens, handling punctuation and filler or stop word like: the, and, a, to ...
  - **Morphological Variants:** Words like “run,” “runs,” and “running” may refer to the same concept but won’t match exactly.
  - **Synonyms:** Different words with the same meaning (“car” vs. “automobile”) will be overlooked.
  - **Context & Semantics:** Token-level matching doesn’t account for context or how words interact to convey meaning.
  - Upper case / lower case letters (e.g., all lower case)
    - Tricky if upper case is required to detect names, grammar
  - Acronyms (U.K. -> UK)
  - Expanding contractions ("don't" -> "do not”).
  - Umlauts (für -> fuer or fuer -> für)
  - Dealing with numbers and symbols in text (“three” -> “3”)
  - Correcting misspelling
- **Text Normalization**: Lowercasing, expanding contractions (e.g., "don’t" → "do not"), handling umlauts (e.g., "fütterung" → "fuetterung") and the other problems mentioned.

> **Handling Tokenization Challenges**
>
> Tokenizing phrases like "Dr. Mayer-Hauser" requires handling abbreviations and compound words. Advanced tokenizers (e.g., spaCy’s) use rules to split such cases accurately.

> **Stop Word Removal Nuances**
>
> While removing stop words improves efficiency, it can lose context (e.g., "to be or not to be" becomes "be not be"). Domain-specific stop word lists may be needed.

Stemming vs. Lemmatization

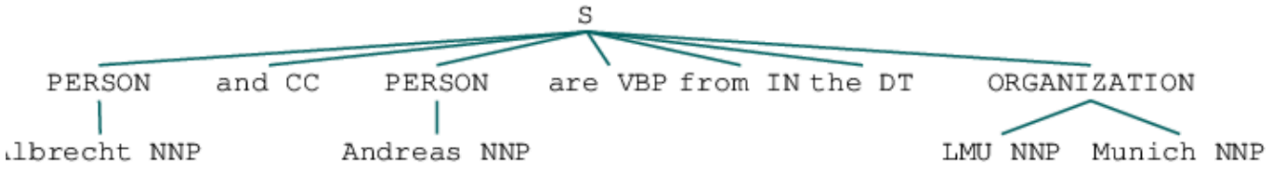
- **Stemming**: Heuristically reducing words to roots (e.g., "running" → "run"). Uses algorithms like Porter Stemmer.
  - Applies crude heuristic rules to strip suffixes (e.g., “running” → “run”)
  - Does not guarantee valid words in the language
  - Faster but can be less accurate

- **Lemmatization**: Linguistically accurate base forms (e.g., "was" → "be"). Relies on dictionaries and grammar rules.
    - Uses vocabulary and morphological analysis (e.g., “running” → “run” if it’s a verb)
        - Produces valid dictionary forms (“run” as a base form)
        - Slower but more linguistically accurate

> **Implementing Stemming/Lemmatization with ML**
>
> Machine Learning can improve stemming/lemmatization by training models on annotated corpora to predict morphological roots, reducing reliance on rule-based systems.

Named Entity Recognition (NER)

```
text = "Albrecht and Andreas are from the LMU Munich."
pos_tags = pos_tag(word_tokenize((text))
named_entities = ne_chunk(pos_tags)
IPython.core.display.display(named_entitied)
```



- Identifies entities (e.g., people, organizations) in text.
- Example: "LMU Munich" → tagged as `ORGANIZATION`.
- Tools: spaCy, Stanford NER.

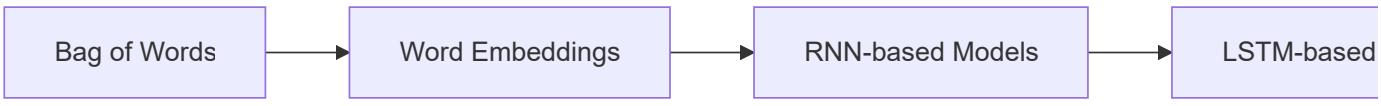
Corpus and Corpora

- **Corpus**: A structured collection of texts (e.g., Project Gutenberg).
- **Parallel Corpora**: Texts in multiple languages (e.g., translated books).

Bag of Words:

The Bag of Words (BoW) model represents text as word frequencies, ignoring grammar and order. While simple, it underpins tasks like spam detection. For example:

- "The cat sleeps. The dog barks." → {"the":2, "cat":1, "sleeps":1, "dog":1, "barks":1}.
  Limitations include missing semantic relationships (e.g., "not good" vs. "good").



Lecture 6 - NLP 2

ELIZA – The Early Chatbot

ELIZA, developed by Joseph Weizenbaum in 1966, is one of the earliest examples of a chatbot that simulates conversation by using pattern matching and substitution techniques.

- **Core Mechanism:** ELIZA analyzes input sentences using decomposition rules triggered by keywords and then generates responses based on reassembly rules.
  - Identification of key words
  - Extraction of minimal context
  - Transforming input into appropriate responses
    ELIZA laid the groundwork for subsequent developments in conversational agents and remains a seminal work in NLP history.

> **ELIZA Technical Description - Weizenbaum (1966)**
>
> "ELIZA is a program [...] which makes certain kinds of natural language conversation between man and computer possible. Input sentences are analyzed on the basis of decomposition rules which are triggered by key words..."

Evolution of NLP Approaches

NLP has undergone significant transformations since its inception:

- **Rule-Based Approaches (1950s–1980s):**
  Early systems relied on handcrafted rules to process language.
- **Statistical Approaches (1980s–2010s):**
  The advent of machine learning introduced probabilistic models that improved language understanding.
- **Deep Learning Approaches (2010s–Present):**
  Neural networks, such as recurrent and transformer-based models, have revolutionized the field with their ability to learn complex patterns from vast datasets.

Approach Type	Characteristics	Example Applications
Rule-Based	Handcrafted rules, deterministic	Early chatbots, simple parsers
Statistical	Probabilistic models, relies on large corpora	Machine translation, tagging
Deep Learning	Neural network architectures, data-intensive	Modern chatbots, sentiment analysis

> **Advantages vs. Disadvantages**
>
> Rule-based systems are interpretable but inflexible, while deep learning models require large datasets and computing power but can generalize better.

Text Analytics: Concepts and Applications

Definition: The use of machine learning and statistical techniques to uncover patterns and trends in textual data.

Text analytics is the process of deriving high-quality information from text.

- **Applications:**
  - Analyzing customer reviews
  - Monitoring social media sentiment
  - Extracting key themes from documents
  - Facilitating automated translation and summarization
- **Core Tasks:**
  - **Language Detection:** Identifying the language of a given text using APIs and libraries.
  - **Named Entity Extraction:** Identifying names of people, places, and organizations.
  - **Sentiment Analysis:** Classifying text as positive, negative, or neutral.
  - **Text Summarization:** Reducing texts to their essential points.

> **Text Analytics Definition**
>
> "Text data mining applies machine learning and statistical methods to texts to discover useful patterns."

Language Identification in Text Analytics

Language identification is fundamental for processing multilingual text:

- **Purpose:** Determines the language of a text to facilitate subsequent processing like translation or sentiment analysis.
- **Tools and APIs:**
  - IBM Language Translator
  - Microsoft Azure Translator
  - Google Translate Language Detection
  - Python libraries (e.g., langdetect, NLTK examples)
- **Process:** The text is analyzed to match linguistic features against known language models.

> **Proper language detection is essential, as misidentification can lead to cascading errors in further NLP tasks.**

Sentiment Analysis

Sentiment analysis aims to determine the emotional tone behind a text:

- **Process:**
  - The text is divided into sentences or phrases.
  - Each segment is analyzed for sentiment using a combination of sentiment libraries and rule-based scoring.
- **Examples:**
  - Classifying a product review as positive or negative.

    - Analyzing social media posts to gauge public opinion.
- **Challenges:**
    - Handling negations (e.g., "not good").
    - Differentiating subtle emotional nuances.

> **Sentiment Analysis with NLTK**
>
> "Sentiment analysis involves breaking text into parts, scoring each, and aggregating these scores for an overall sentiment."

> **VADER - Hutto & Gilbert (2014)**
>
> "VADER is a parsimonious rule-based model for sentiment analysis of social media text."

Text Summarization Techniques

Text summarization reduces a large text into a concise version that retains the key information:

- **Two main approaches:**
    - **Extraction:** Directly selecting important sentences from the text.
    - **Abstraction:** Generating a new summary that conveys the critical content in a rephrased manner.

> **Extraction vs. Abstraction**
>
> Extraction is simpler as it selects verbatim sentences, whereas abstraction requires deeper semantic understanding to generate new text.

- **Workflow Example:**
    1. **Sentence Segmentation:** Convert paragraphs into sentences.
    2. **Preprocessing:** Clean and normalize the text.
    3. **Tokenization:** Break sentences into words.
    4. **Frequency Analysis:** Identify weighted word frequencies.
    5. **Sentence Scoring:** Replace words with frequency scores and rank sentences.
    6. **Summary Generation:** Combine top-ranked sentences into a final summary.

> **Summarization Example**
>
> "Text summarization extracts the most relevant parts of a text to create a concise version that still conveys the original message."

Preprocessing Techniques in NLP

Effective text analysis relies on several core preprocessing steps:

- **Tokenization:** Breaking text into words or sentences.
- **Stop Words Removal:** Eliminating common words that add little semantic value.
- **Normalization:** Converting text to a uniform format (e.g., lowercasing, removing punctuation).
- **Stemming and Lemmatization:** Reducing words to their base or root form.
- **Part-of-Speech Tagging:** Identifying the grammatical roles of words.
- **Named Entity Extraction:** Detecting and classifying key entities such as names, locations, and dates.

How do preprocessing steps improve the effectiveness of NLP applications?

Preprocessing removes noise and standardizes text, thereby enhancing the accuracy and efficiency of subsequent analyses.

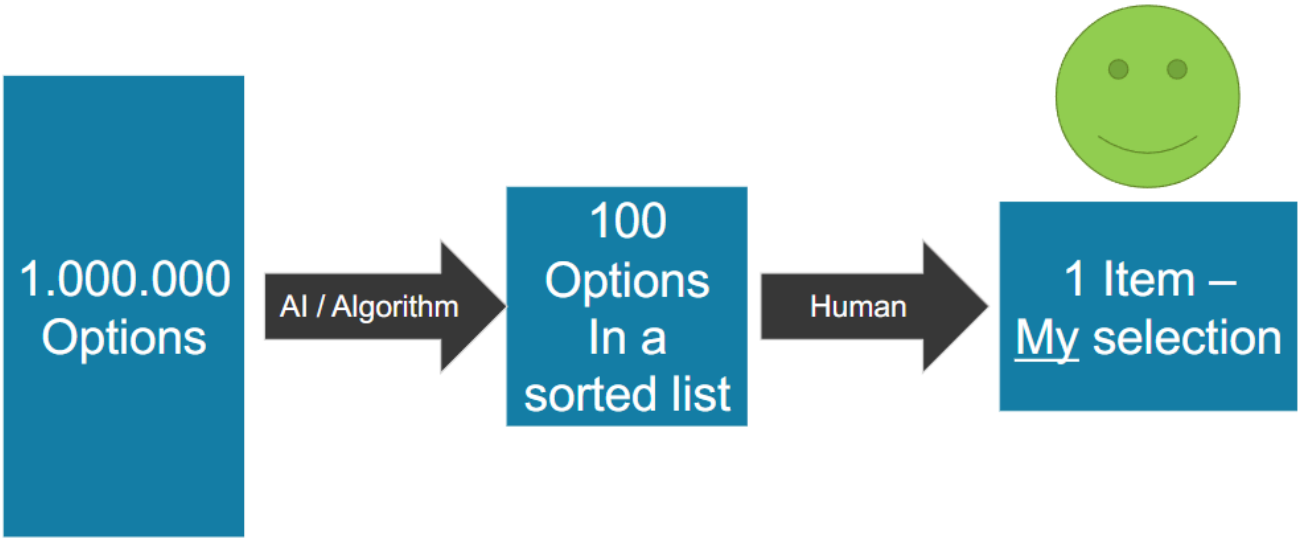
Lecture 7 - Recommender Systems

Recommender Systems (RSs) help users discover relevant content by providing personalized suggestions.

> **Definition of Recommender Systems - Ricci, Rokach & Shapira (2011)**
>
> "Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user. […] The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read."

Why Are Recommender Systems Needed?

- The explosion of digital content makes **manual selection impractical**.
- AI-generated content further increases the **amount of available information**.
- Without RSs, users struggle to **find relevant content efficiently**.



> **Why Random Selection Fails**
>
> Most digital content is **irrelevant** to a particular user. Without intelligent filtering, random selection leads to a **frustrating user experience**.

Types of Recommender Systems

Content-Based Filtering

- Items are recommended **based on features** they share with previously liked items.
- Uses metadata like **genre, author, keywords, length, or language**.
- Requires a **taxonomy or vector-based representation** of item features.

Collaborative Filtering

- Recommends items **based on user behavior and preferences**.
- Two main types:
  - **User-based:** Finds users with similar preferences.
  - **Item-based:** Finds items that receive similar ratings.
- Requires a **large dataset of user interactions**.

Difference Between Content-Based and Collaborative Filtering

Content-based filtering analyzes **item features**, while collaborative filtering relies on **user interactions and similarities**.

Hybrid Approaches

- Combines **content-based and collaborative** filtering techniques.
- Helps address **cold-start problems**, where user data is initially unavailable.
- Example: **Netflix uses hybrid models** to improve recommendations.

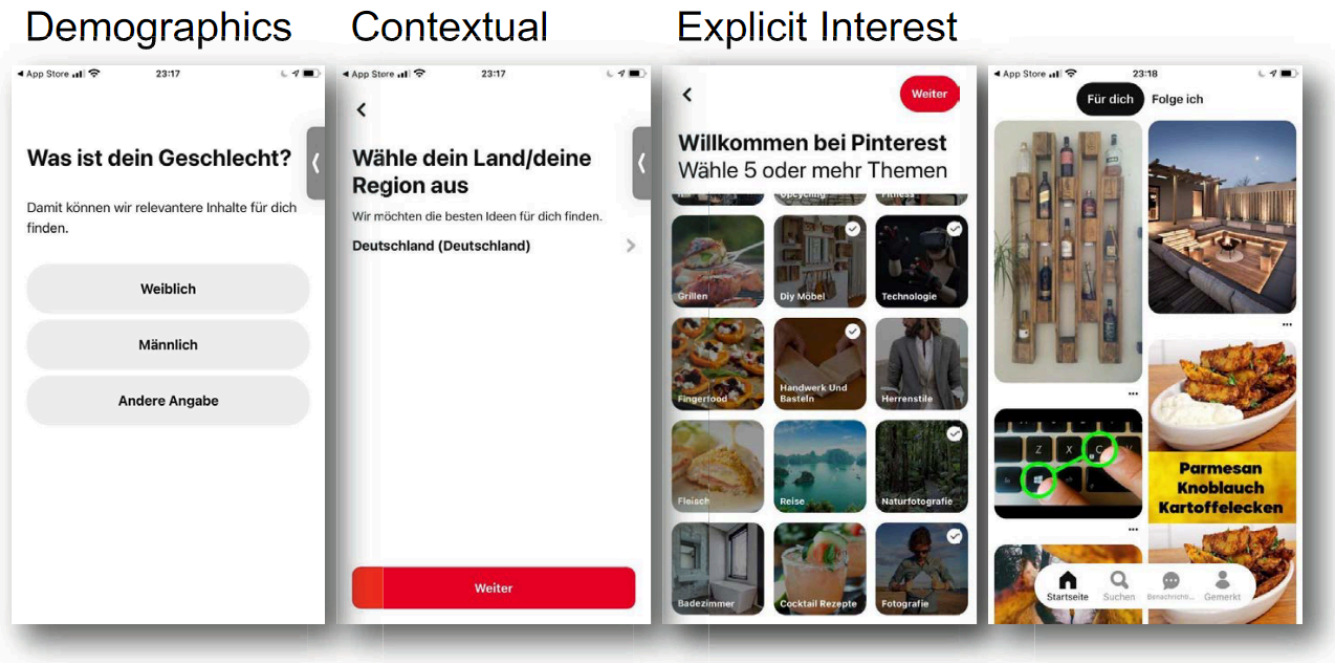
Type	Basis for Recommendation	Pros	Cons
Content-Based	Item characteristics	Works well with known items	Needs rich item metadata
Collaborative	User behavior	Learns from others' actions	Requires large user data
Hybrid	Both	Balances both approaches	Computationally expensive

---

Challenges in Recommender Systems

Cold-Start Problem

- Occurs when **new users or items** have no prior data.
- Solutions:
  - Asking users for initial preferences.
  - Using **demographic data** or pre-trained models.
  - Hybrid filtering to start with content-based and later transition to collaborative filtering.



Sparsity of Ratings

- Many users **rate only a small number of items**, making data sparse.
- Solution: Matrix factorization techniques (e.g., **SVD, ALS**) to infer missing ratings.

Bias and Filter Bubbles

- RSs may **reinforce existing preferences** and limit diversity.
- Solution: Introduce **serendipity and exploration factors** into recommendations.

> **Importance of Serendipity**
>
> Users often **enjoy discovering unexpected content** that aligns with their interests but isn't an obvious choice.

User Interface (UI) and User Experience (UX) Considerations

Transparency and Control

- Users prefer to **understand why** an item is recommended.
- RSs should allow users to **adjust preferences and provide feedback**.

Example:

- **Version 1:** "Take the train at 12:17."
- **Version 2:** "Do you prefer the **12:17 train (45 min)** or the **12:15 bus (50 min, unreliable)?"

Reducing Cognitive Load

- The **number of choices** should match the UI context.
- Example:
  - **Voice UI:** Few choices (1-3 options).
  - **Mobile app:** Moderate number of choices (5-10 items).
  - **Desktop UI:** Large selection possible (20+ items).

> **Why UI Matters**
>
> A well-designed UI makes RSs feel **helpful rather than intrusive**. If users feel **forced**, they might reject recommendations altogether.

Machine Learning in Recommender Systems

Similarity Metrics

- Used in collaborative filtering to measure user/item relationships:
  - **Cosine similarity**
  - **Pearson correlation**
  - **Euclidean distance**

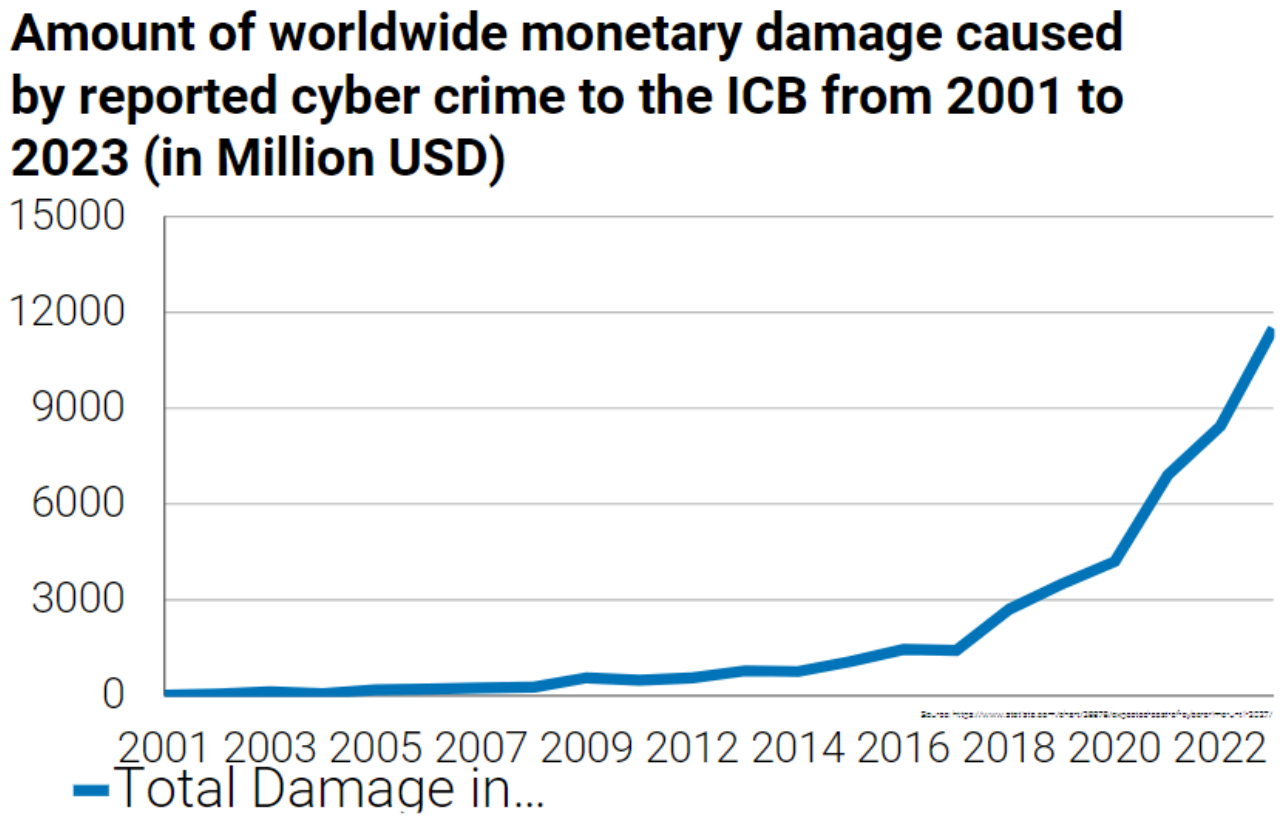
Matrix Factorization Techniques

- **Singular Value Decomposition (SVD)** – Breaks down user-item interactions into latent factors.
- **Alternating Least Squares (ALS)** – Optimizes for missing data in sparse matrices.
- **Neural Networks & Deep Learning** – Used for large-scale recommendation tasks.

> **How Machine Learning Helps RSs**
>
> ML allows RSs to **predict user preferences**, optimize for **personalization**, and improve over time **based on feedback**.

Lecture 8 - Security and Privacy

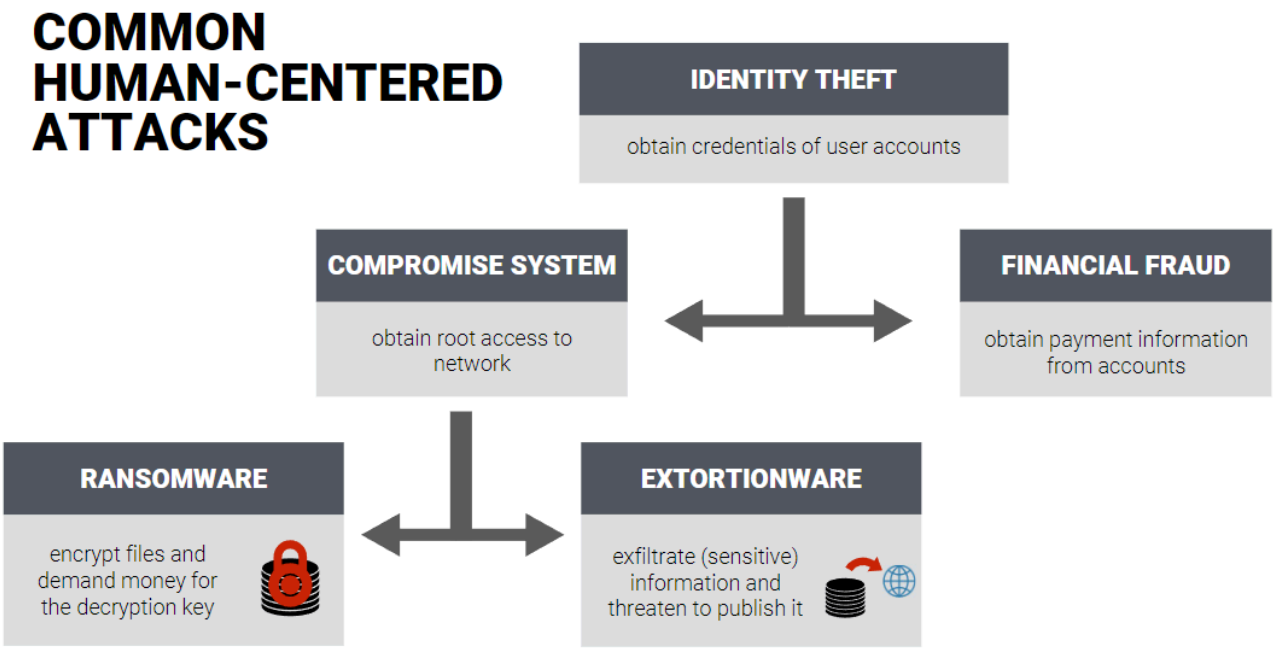
The Growing Cost of Cybercrime

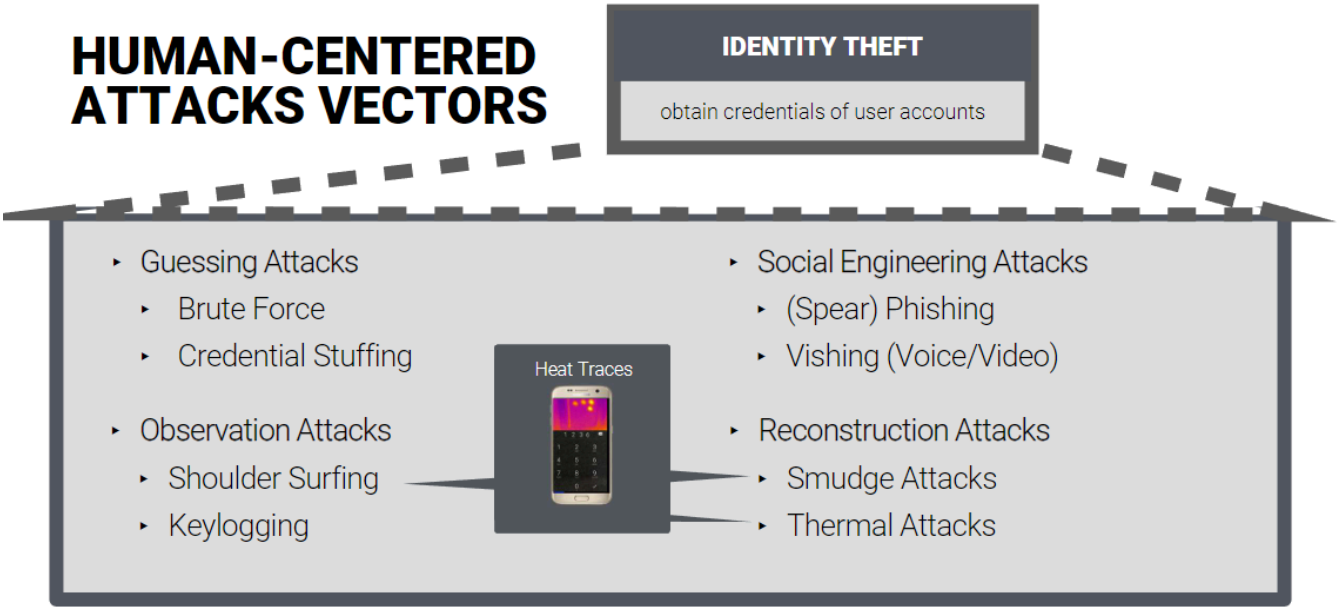


- Cybercrime costs have increased exponentially, reaching **billions of dollars annually**.
- Successful **cyberattacks on German universities** in recent years highlight the vulnerabilities of institutions.

> **Cybercrime Damage Trends - Statista**
>
> "The financial impact of cybercrime has grown significantly, with expected damages surpassing $10 trillion by 2025."

> **On Hackers, TK Keanini (CISCO)**
>
> "Hackers don't break in — They log in"





Common Human-Centered Attack Vectors

Attack Type	Description	Example Scenarios
Identity Theft	Stealing credentials to access user accounts	Phishing, credential stuffing
Extortionware	Exfiltrating sensitive data and threatening exposure	Ransom demands
Ransomware	Encrypting files and demanding payment for decryption	Corporate attacks
System Compromise	Gaining root access to a network	Privilege escalation
Financial Fraud	Obtaining payment details for unauthorized transactions	Credit card fraud

How to Prevent Phishing

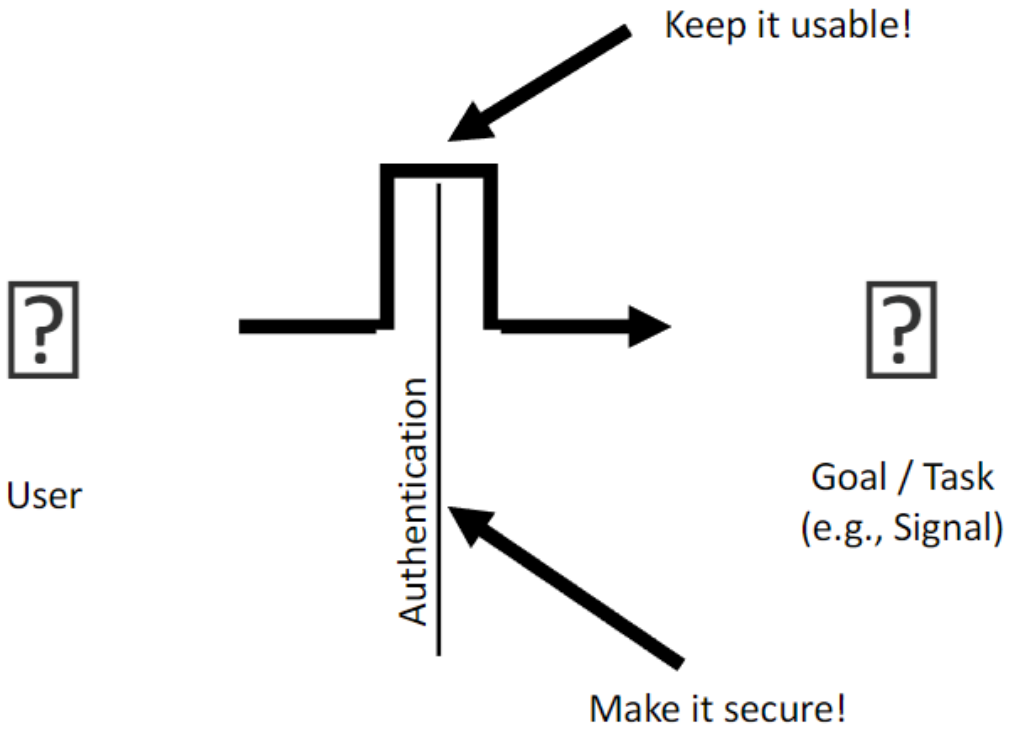
Never click on unexpected email links. Verify sender authenticity and enable multi-factor authentication.

Human-Centered Security Challenges

Security vs. Usability Trade-Off

- Security is often a **secondary task** for users, meaning convenience takes priority.
- **Users want quick access**, while security mechanisms aim to restrict unauthorized entry.
- A study by **Adams et al. (1999)** shows this misalignment between users and security experts.

> **Security vs. Usability - Adams et al. (1999)**
>
> "If security mechanisms are not usable, they are not secure."



Frequent Authentication Weaknesses

- **Common PINs & Passwords**: Users often pick predictable credentials.
- **Shoulder Surfing & Smudge Attacks**: Observers can reconstruct passwords from traces left on screens.
- **Brute Force & Credential Stuffing**: Automated attempts to guess passwords.
- **Social Engineering**: Manipulating users into revealing credentials.

> **Why Password Complexity Fails**
>
> Users who are forced to create complex passwords **often write them down**, reducing security effectiveness.

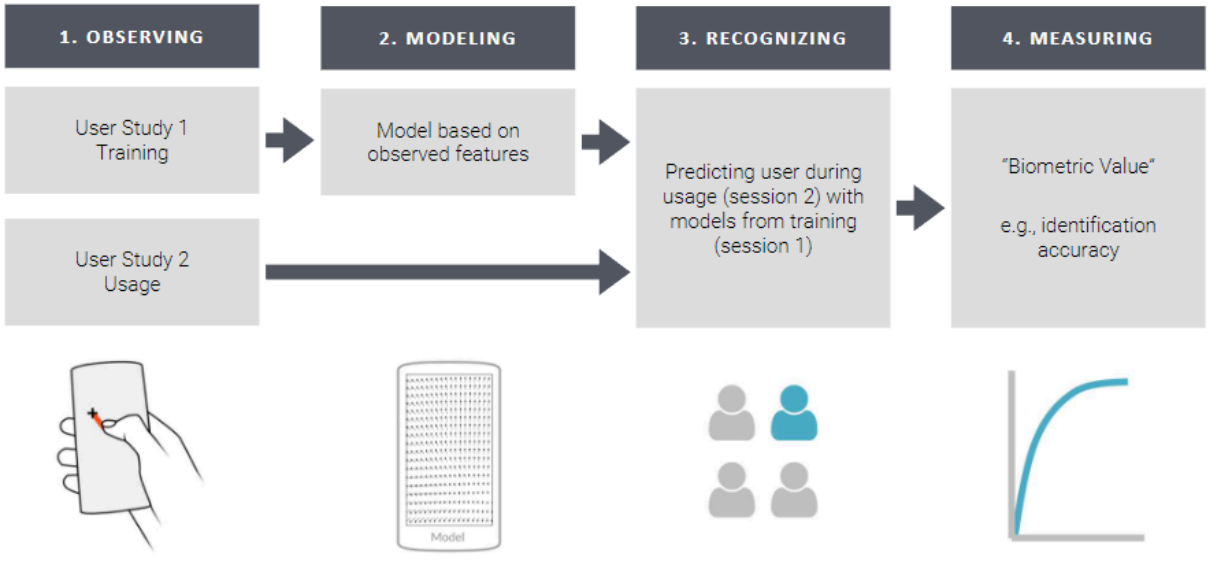
Security	Usability / HCI	Usable Security
Humans are a secondary constraint to security constraints	Humans are the primary constraint, security rarely considered	Human factors and security are both primary constraints
Humans considered primarily in their role as adversaries / attackers	Concerned about human error but not human attackers	Concerned about both normal users and adversaries
Involves threat models	Involves task models, mental models, cognitive models	Involves threat models AND task models, mental models, etc.
Focus on security metrics	Focus on usability metrics	Considers usability and security metrics together
User studies rarely done	User studies common	User studies common, often involve deception + active adversary

---

Authentication Methods

Behavioral Biometrics

- Users can be identified based on **unique behavioral traits**, such as:
  - **Typing biometrics** (e.g., flight time, hold time)
  - **Gait recognition** (e.g., walking patterns)
  - **Interaction data** (e.g., app usage, touch dynamics)



> 🔍 **Behavioral Biometrics:**
> Unlike passwords, behavioral biometrics continuously authenticate users **without requiring explicit actions**.

Intelligent Authentication Systems

- Security interfaces are integrating **machine learning** to detect anomalies.

- Continuous authentication monitors user behavior to **detect unauthorized access**.
- Example: A system detects unusual typing speed and flags potential impostors.

Privacy Concerns & Transparency

Factors Influencing Privacy Perception

Privacy Concern	Explanation
Device Type	Smart home devices raise more concerns than personal computing devices.
Social Context	People are more comfortable sharing data with close contacts.
Location Sensitivity	Privacy concerns increase in intimate settings (e.g., bedrooms).

> **Privacy vs. Convenience**
>
> Users will trade **privacy for convenience** if benefits outweigh perceived risks.

Transparency & User Control

- Providing **privacy dashboards** enhances user trust.
- Field studies show that users **rarely engage with fine-grained controls** unless actively encouraged.
- Transparency without control **may increase privacy concerns** rather than reduce them.

> **Transparency vs. Control - MobileHCI 22**
>
> "Users prefer having control, but they rarely use it unless directly prompted."

Designing Usable Security & Privacy Interfaces

Balancing Security & Usability

Approach	Benefit	Challenge
Two-Factor Authentication	Adds an extra layer of security	Inconvenient for frequent logins
Password Managers	Reduces password fatigue	Users must trust the tool
Biometric Authentication	Quick & user-friendly	Privacy concerns over data storage

Fine-Grained Privacy Controls

- Traditional **binary permission models** (Allow/Deny) are **too rigid**.
- **Privacy Sliders** allow users to **adjust data sharing levels** dynamically.

> **Privacy Sliders:**
> Instead of all-or-nothing permissions, sliders enable users to fine-tune access **based on context and**

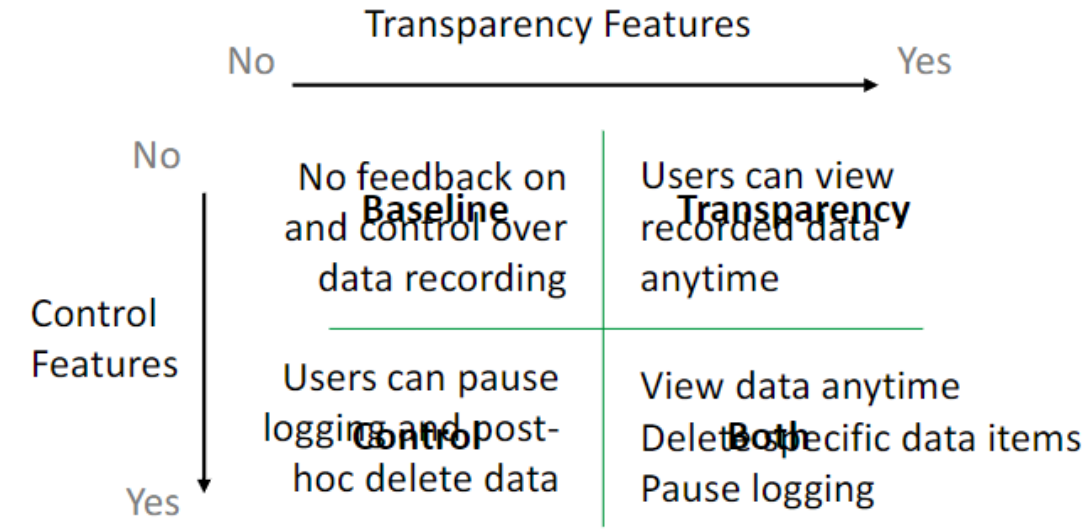
> **necessity**.

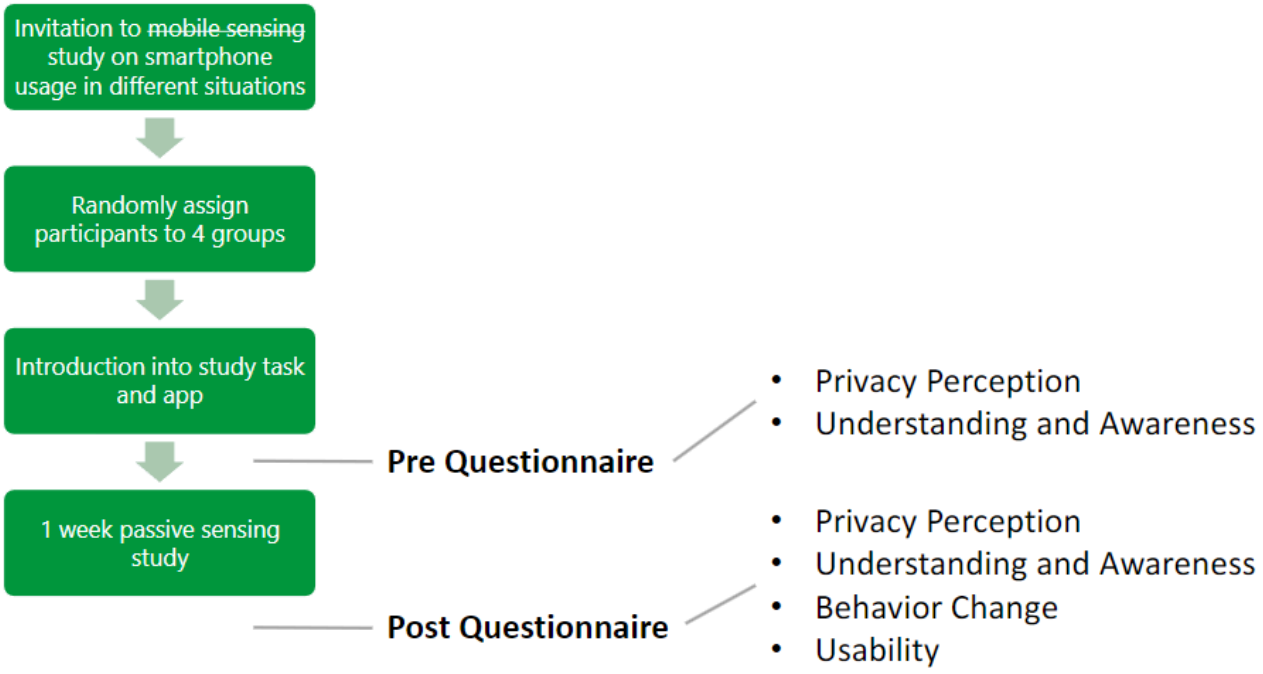
Field Study Insights on Security UI Adoption

- **Users prioritize ease of access over security settings.**
- **Warning fatigue** reduces the effectiveness of security prompts.
- **Default settings** strongly influence user behavior.

> **How to Reduce Warning Fatigue**
>
> Use **adaptive security prompts** that trigger only under high-risk scenarios.

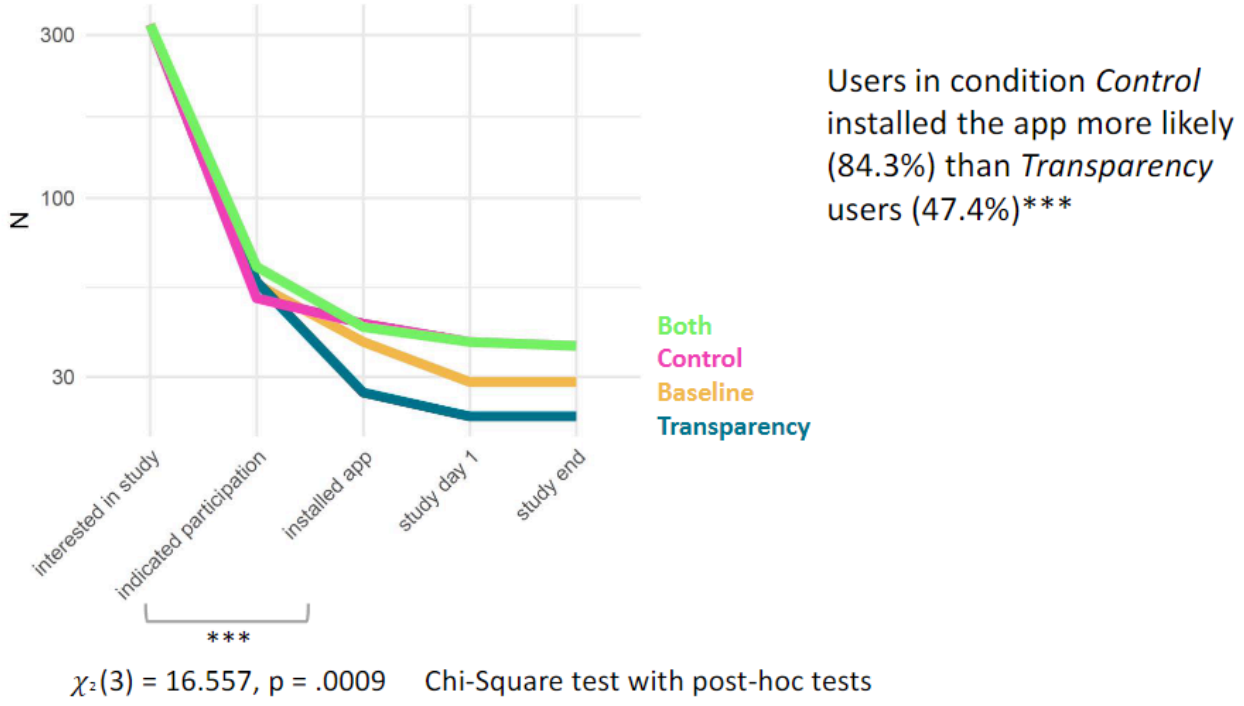
Transparency and Control in Mobile Sensing Apps





- **Field Study Design:** A 2x2 factorial design study compared four conditions: baseline, transparency only, control only, and both transparency and control.

- **Findings:** Users provided with transparency features initially showed higher understanding of data logging. However, transparency alone did not significantly improve adoption rates; control features are critical to enhancing user trust.
- **User Behavior:** Despite the availability of control mechanisms, users tend to use them sparingly, indicating that such features should be intuitive and accessible on-demand.



- **Understanding**- What happens with my data?
  - In the beginning, Transparency users had higher understanding than others
  - Understanding of Transparency users decreased, while the others' increased
- Awareness – What data is logged?
  - Slightly higher knowledge about what is logged for Transparency users
  - Slight improvements during the study period

> **Importance of Combining Transparency with Control**
>
> Ensure that any transparency feature is paired with an easy-to-use control mechanism to help users feel secure without overwhelming them.

Fine-Grain Privacy Control and the Privacy Slider Concept

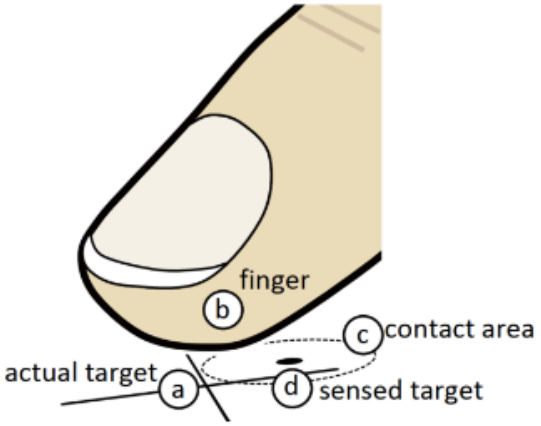
- **Limitations of Binary Permissions:** Traditional permission systems force users into yes/no decisions, which can be overly restrictive or too permissive.
- **Privacy Slider Concept:**
  - A privacy slider allows users to adjust their data sharing preferences along a continuum rather than making binary choices.
  - Studies indicate that users prefer interfaces that mirror the natural structure of the data (e.g., sliders for continuous values).
- **User Feedback:** While the privacy slider enhances perceived control and transparency, designers must be wary of warning fatigue, where too many alerts can lead to desensitization.

Lecture 9 - Text Entry

Mobile Text Entry

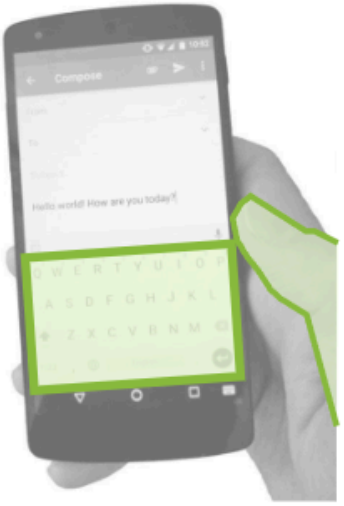
Why is typing on mobile devices difficult?

Parallax
eye – finger - screen



[Holz and Baudisch 2011]

Mobile use
1-2 fingers, small keys, body movement



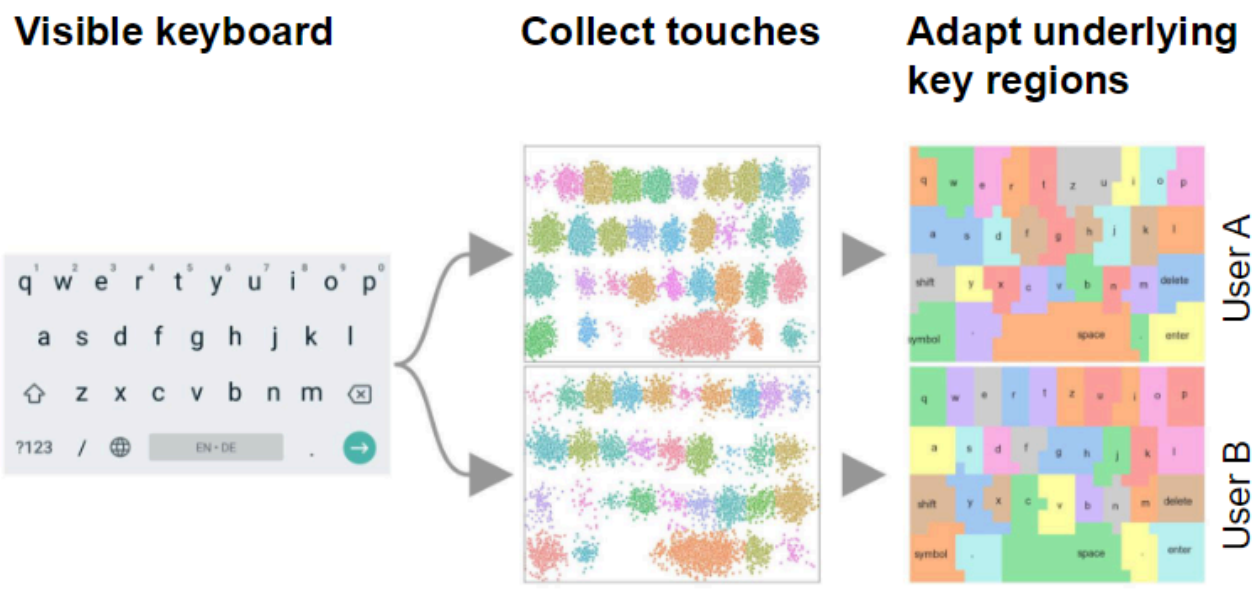
- **Parallax effect:** Misalignment between finger, screen, and eyes leads to inaccurate touches.
- **Small key sizes:** Higher chance of pressing the wrong key.
- **Mobile use conditions:** Users type in various postures (one-handed, walking, sitting), affecting accuracy.

> **Parallax and Typing Accuracy - Holz & Baudisch (2011)**
>
> "Parallax between the user's eye, finger, and screen affects the precision of touch input."

Touchscreen Keypress Variability

- Studies show that users **do not touch keys at their geometric centers**.
- Typing patterns vary by individual and context (e.g., smartphone vs. tablet).
- Gaussian distributions can model touch location variation.

Probabilistic Keyboard Models



- **Touch points follow a probabilistic distribution** around key centers.
- **Bayes' Rule** is used to infer the most likely intended key.

> **What is a Probabilistic Keyboard Model?**
>
> Instead of treating touches as exact, probabilistic models **assign likelihoods** to different keys based on touch input and language context.

Keyboard Adaptation Strategies

User-Specific Adaptations

Adaptation Type	Description
Touch Behavior	Adjusts key regions based on user's past typing patterns.
Context Awareness	Changes layout based on hand posture or movement (e.g., walking).
Language Model	Predicts likely words based on previous input.

Gaussian Key Model

- Each key is modeled as a **Gaussian distribution**.
- The system updates key regions dynamically based on **historical touch data**.

Error Correction & Prediction

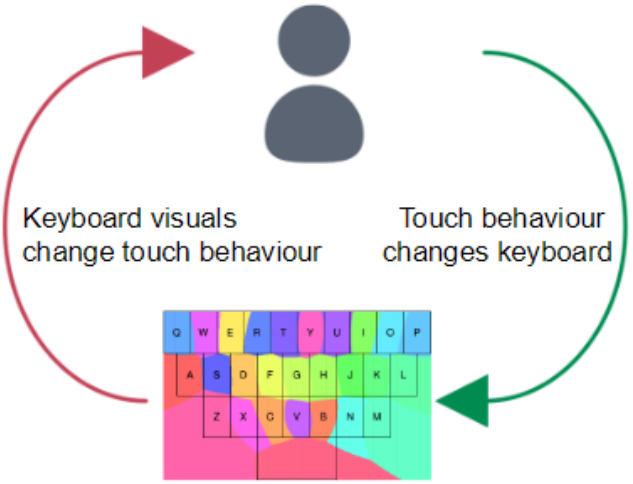
Language Model Influence

- After a key is detected, the system applies **language models (e.g., bigram models, neural networks)** to improve accuracy.
- Example: If a user types "th", the system predicts "the" over "thu".

Why do our keyboards not look like this?

[Yin et al. 2013]

→ Avoid co-adaptation of user and system



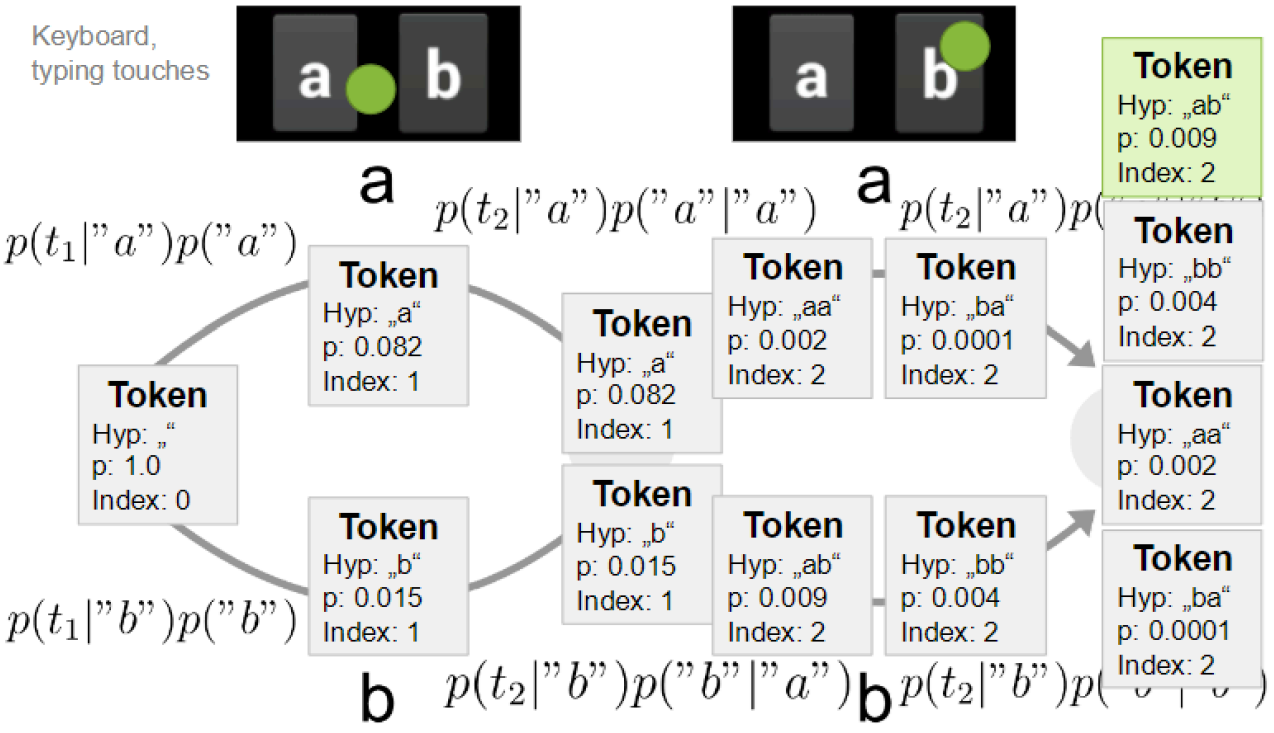
How Language Models Improve Typing

Even if a touch input is slightly off, language models help select the **most likely** intended word.

Decoding Typing Sequences

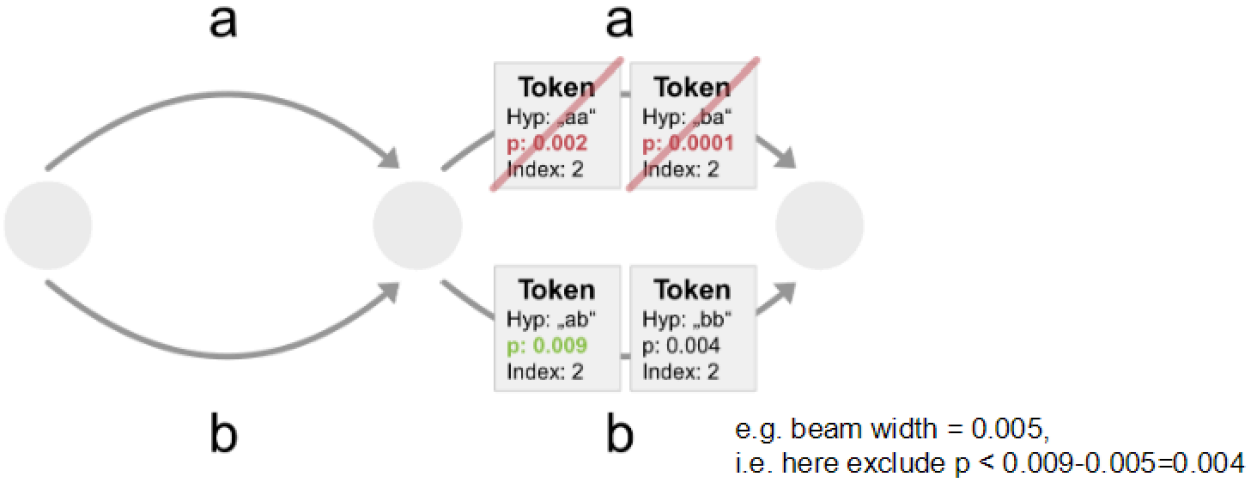
Method	Description
Token Passing	Iteratively refines possible words based on touch input.
Beam Search	Limits search space to the most probable word sequences.

Token passing algorithm



With beam search / pruning

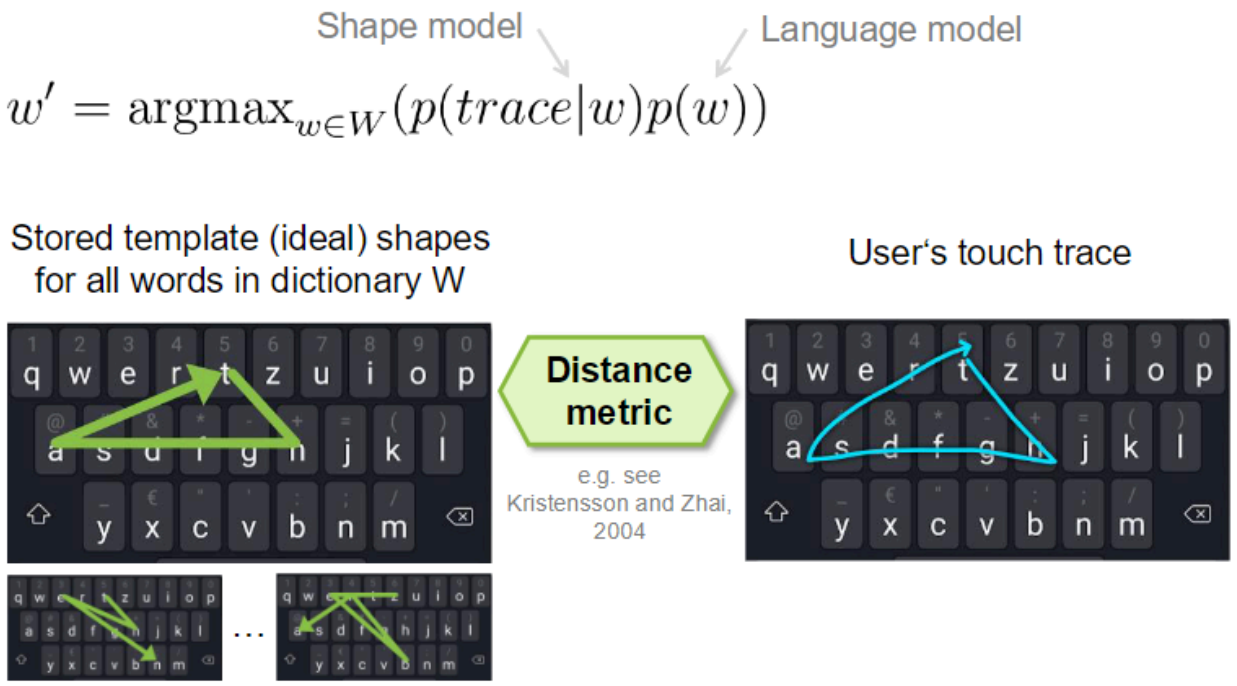
- Problem: Large search space
  Substitution-only → exponential, Insertion → infinite
- Solution: **Beam search / pruning**
  Per index, only propagate tokens that are within a certain range (=„beam width“) of the probability of the most likely token.



Gesture-Based Typing

- Swiping from letter to letter forms a trace that is matched to stored **word shapes**.
- Uses **distance metrics** to compare user input with predefined templates.

- Infer intended word from shape of finger trace on the keyboard



> Gesture-Based Typing - Kristensson & Zhai (2004)
>
> "Users can enter text by drawing word shapes, reducing individual key presses."

Optimization-Based Keyboard Design

Why Optimize Keyboard Layouts?

- QWERTY is **historically optimized for typewriters**, not touchscreens.
- Alternative layouts reduce **finger movement** and improve efficiency.

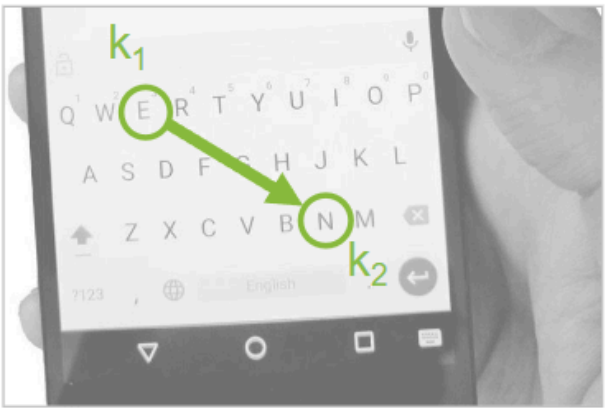
Optimization Strategies

Method	Description
Random Search	Generates random layouts and selects the best.
Simulated Annealing	Gradually refines layouts by accepting probabilistic changes.
Heuristic Methods	Use AI models to find the best design based on constraints.

Objective Function: How to judge a layout?

- Finger movement time (e.g. Fitts' law)

$$t(k_1, k_2) = a + b \log_2 \left(\frac{D}{W} + 1\right)$$



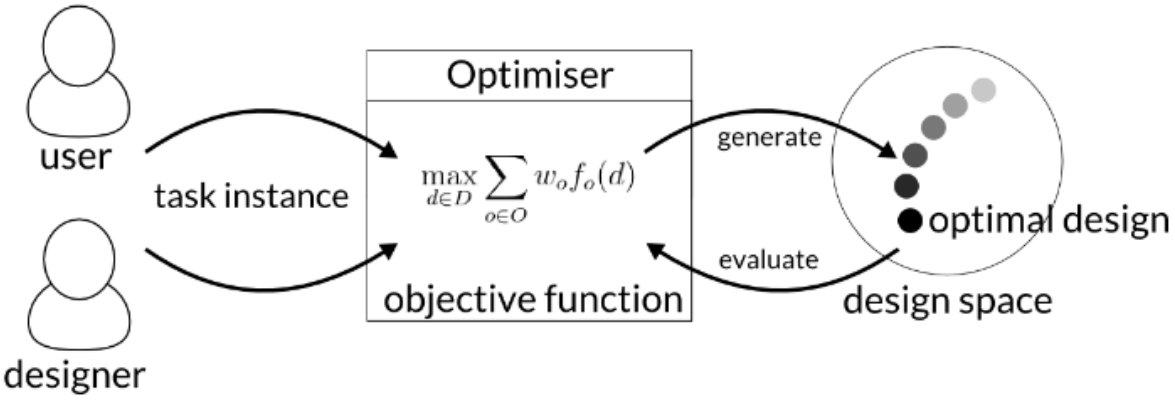
- Language properties (e.g. bigram frequencies)

e.g. $p(\text{"n"} \mid \text{"e"}) = 0.001$

- Combined: mean time between two key presses

$$f(d) = \sum_{k_1 \in K} \sum_{k_2 \in K} p(d(k_2) | d(k_1)) t(k_1, k_2)$$

where the design d maps from keys to characters



Simulated Annealing Example

- **Starts with a random layout**.
- Gradually refines by testing **small changes**.
- "Cools down" to **converge on an optimal design**.

Potential of Optimization-based Design

- Obtaining information on the design problem and a formal specification
- Exploring a large design space comprehensively
- Improving quality and robustness of designs
- Estimating possible improvements
- Supporting human designers
- Optimization during use, personalised UIs
- Requires: Models of user behaviour, formal problem definition / objective function, computational capacity, …

Why is QWERTY Still Used?

Even though optimized layouts exist, users are **resistant to change** due to familiarity.

Questions & Answers

Name and explain the key components of optimization-based UI design.

Key components include **objective functions** (define what is being optimized), **design constraints** (limit solutions to practical options), and **search algorithms** (find the best possible solution).

How can designers influence obtained designs in this approach?

Designers can adjust **constraints, weighting factors, and optimization goals** to guide the system towards practical and user-friendly solutions.

Explain Simulated Annealing. Can you consider a design resulting from this method as "optimal"? Why (not)?

Simulated Annealing is an optimization technique that explores different solutions by gradually reducing randomness in search. The result is **near-optimal**, but not guaranteed to be the absolute best.

If it is possible to find better designs than QWERTY, why are we not using them widely?

User habits, resistance to change, and infrastructure dependence make it difficult to transition away from QWERTY despite superior alternatives.

Beyond keyboard layouts, which other UI design problems could be addressed with this approach? And which are hard to address in this way?

Optimization can improve **menu layouts, gesture-based interfaces, and adaptive UI designs**. However, **social interaction and cultural usability** are harder to optimize due to subjective factors.

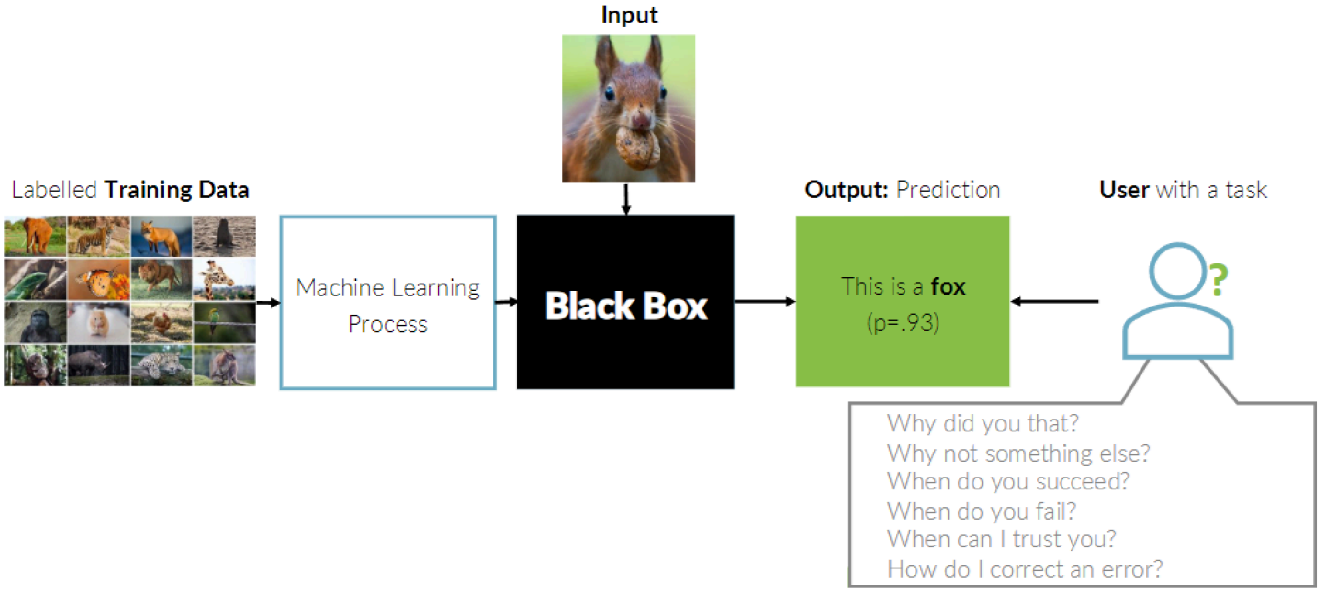
Lecture 10 - Explainable AI

Instead of accepting opaque “black box” outputs, Explainable AI (XAI) strives to provide insights into how decisions are made. This not only increases user trust but also supports error diagnosis, bias detection, and regulatory compliance.

> Yudkowsky 2008
>
> "By far the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."

The Black Box Problem and the Clever Hans Phenomenon

Modern machine learning models often operate as black boxes—their internal decision processes remain hidden from end users. This opacity can lead to misinterpretations, as exemplified by the “Clever Hans” problem, where a model might appear to understand a task while actually exploiting spurious patterns.



> 🔎 **Black Box Problem**
> Refers to the challenge of interpreting complex algorithms whose internal logic is not directly accessible or understandable, making it difficult to predict how and why specific decisions are made.

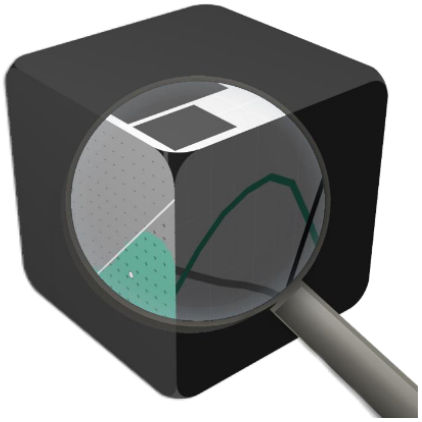
There has always been proprietary, non-interpretable knowledge. What is different now?

- **Scale and Impact:** Modern ML models operate at a vast scale and are embedded in critical systems (e.g., healthcare, finance, autonomous vehicles), making their opaque decision processes have widespread societal consequences.
- **Need for Accountability:** Unlike traditional proprietary knowledge that was confined to specific domains, today’s data-driven algorithms must be interpretable to ensure fairness, prevent bias, and allow for regulatory oversight.

> ***We do not need to understand how a motor works to drive a car – why do we need to understand ML models now?***
>
> - **Dynamic and Uncertain Behavior:** While the mechanics of a motor are stable and predictable, ML models are probabilistic and can evolve with new data, leading to unexpected behavior that impacts decision-making.
> - **High-Stakes Decisions:** ML models often underpin systems that directly affect human lives and societal functions, so understanding their inner workings is crucial for ensuring transparency, trust, and proper accountability.

What is Explanability?

"Explainability," "Interpretability," and "Transparency" are often used interchangeably.



Possible Definitions:

- "… the ability to explain or to present in understandable terms to a human" [Doshi-Velez & Kim 2017]

- "… is the degree to which a human can understand the cause of a decision" [Miller 2017]

- "… is the degree to which a human can consistently predict the model's result" [Kim et al. 2016]

Societal Challenges and Ethical Implications

In sensitive applications like judicial systems, finance, or recruitment, lack of transparency can result in biased outcomes and erode public trust.
GDPR's right to explanation, mandate that automated decisions include human-understandable justifications.

> **Balancing Innovation and Accountability**
>
> AI systems must not only be high-performing but also transparent and fair, ensuring that users are not subjected to unexplained or biased decisions.

What Constitutes a Good Explanation?

Good explanations in AI should:

- **Clarify the decision process:** Helping users understand why a particular output was generated.
- **Build trust:** Enabling users to calibrate their confidence in the system.

- **Support user intervention:** Allowing users to correct or challenge decisions.
- **Be concise yet comprehensive:** Striking the right balance between detail and understandability.

Researchers have offered various definitions:

- “The ability to explain or to present in understandable terms to a human.” ([Doshi-Velez & Kim 2017])
- “The degree to which a human can understand the cause of a decision.” ([Miller 2017])

Interpretability in Machine Learning

Interpretability refers to how well a human can comprehend the reasoning behind a model's output. Two key dimensions are:

Local vs. Global Interpretability

Scope	Focus	Key Question
Local	Individual predictions	"Why did this specific decision occur?"
Global	Overall model behavior	"How does the model generally operate?"

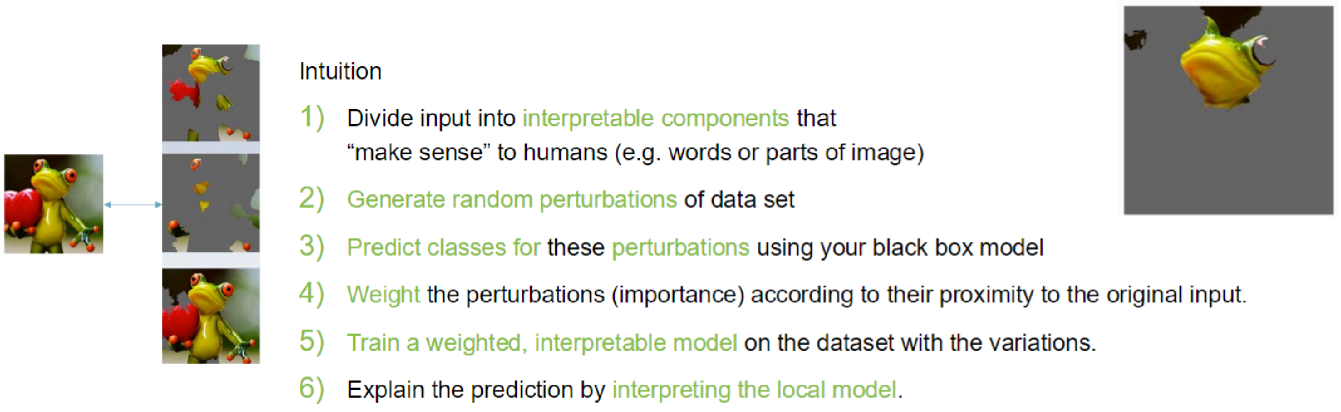
Local explanations help users understand individual outcomes, while global explanations reveal the overarching mechanics of a model.

Intrinsic vs. Post-hoc Interpretability

Interpretability Type	Description	Example Methods
Intrinsic	Models designed to be interpretable by nature	Decision Trees, Linear Models
Post-hoc	Techniques applied after model training to extract explanations	LIME, SHAP

Intrinsic methods build interpretability into the model structure, whereas post-hoc methods generate explanations for otherwise opaque models.

Explanation Methods: Local Interpretable Model-Agnostic Explanations (LIME)



LIME is a popular post-hoc method that provides local interpretability.

1. **Decomposition:** Dividing an input into interpretable components (e.g., words or image segments).
2. **Perturbation:** Generating variations of the input data.
3. **Prediction:** Using the original black box model to predict outcomes for these perturbations.
4. **Weighting:** Assigning importance based on the similarity to the original input.
5. **Local Modeling:** Training a simple, interpretable model on the weighted perturbed data.
6. **Explanation:** Presenting the factors that most influenced the model's prediction.

Applications of Explainability

Explainability in AI serves multiple functions:

- **Model Validation:** Detecting and eliminating bias in training data.
- **Model Debugging:** Identifying reasons behind misclassifications or errors, including adversarial attacks.
- **Knowledge Discovery:** Uncovering new insights from data by revealing hidden patterns and correlations.

Each of these applications contributes to a more reliable and accountable AI system, benefiting both developers and end users.

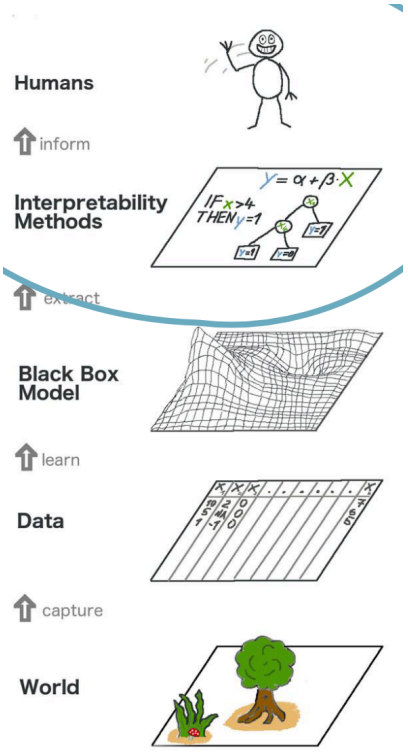
Human-Centered AI and HCI Challenges

Human-Centered Artificial Intelligence (HCAI) prioritizes enhancing human performance and ensuring that systems are safe, reliable, and trustworthy. Key challenges include:

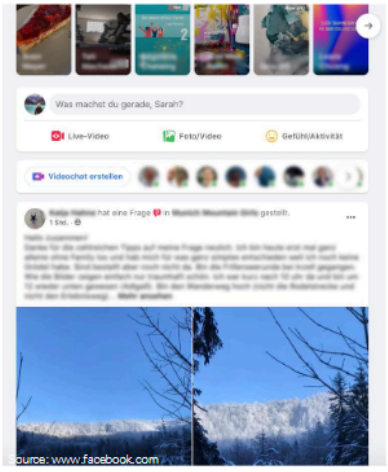
- **Understanding:** Helping users build accurate mental models of how AI systems operate.

- **Trust:** Enabling users to gauge when to rely on the system and when to question its outputs.

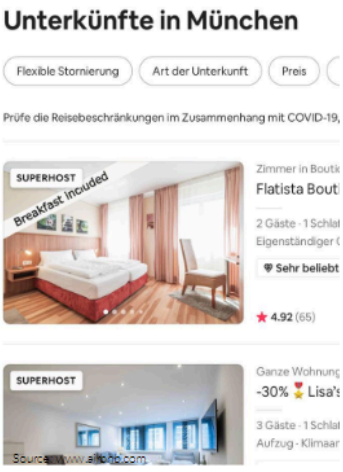
- **Control:** Allowing users to provide feedback and correct system errors.



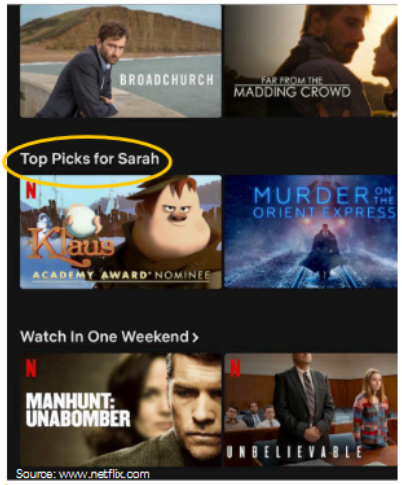
Source: [Molnar 2019]



Lack of Algorithmic Awareness



Algorithmic Anxiety



Intransparent Recommendations

Enhancing User Trust

Clear, transparent explanations empower users to engage confidently with AI systems and mitigate algorithmic anxiety.

Support Strategies and User Control

Real-world systems, such as recommendation engines or navigation apps, often suffer from issues like:

- **Lack of Feedback:** Users have few opportunities to inform the system of errors or misinterpretations.
- **Limited User Control:** Systems may override user preferences without clear justification.

- **Intransparent Recommendations:** Explanations that fail to reveal the reasoning behind suggestions.

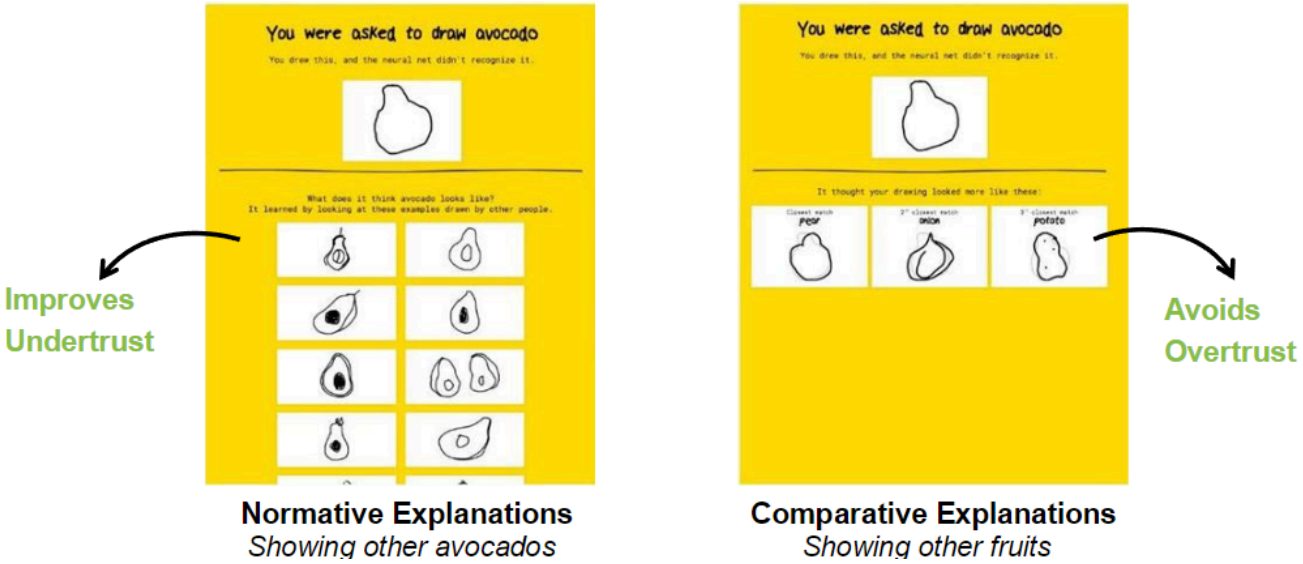
Improving these areas requires designing interfaces that not only provide explanations but also invite interactive user feedback and correction.

Discussion and Future Directions

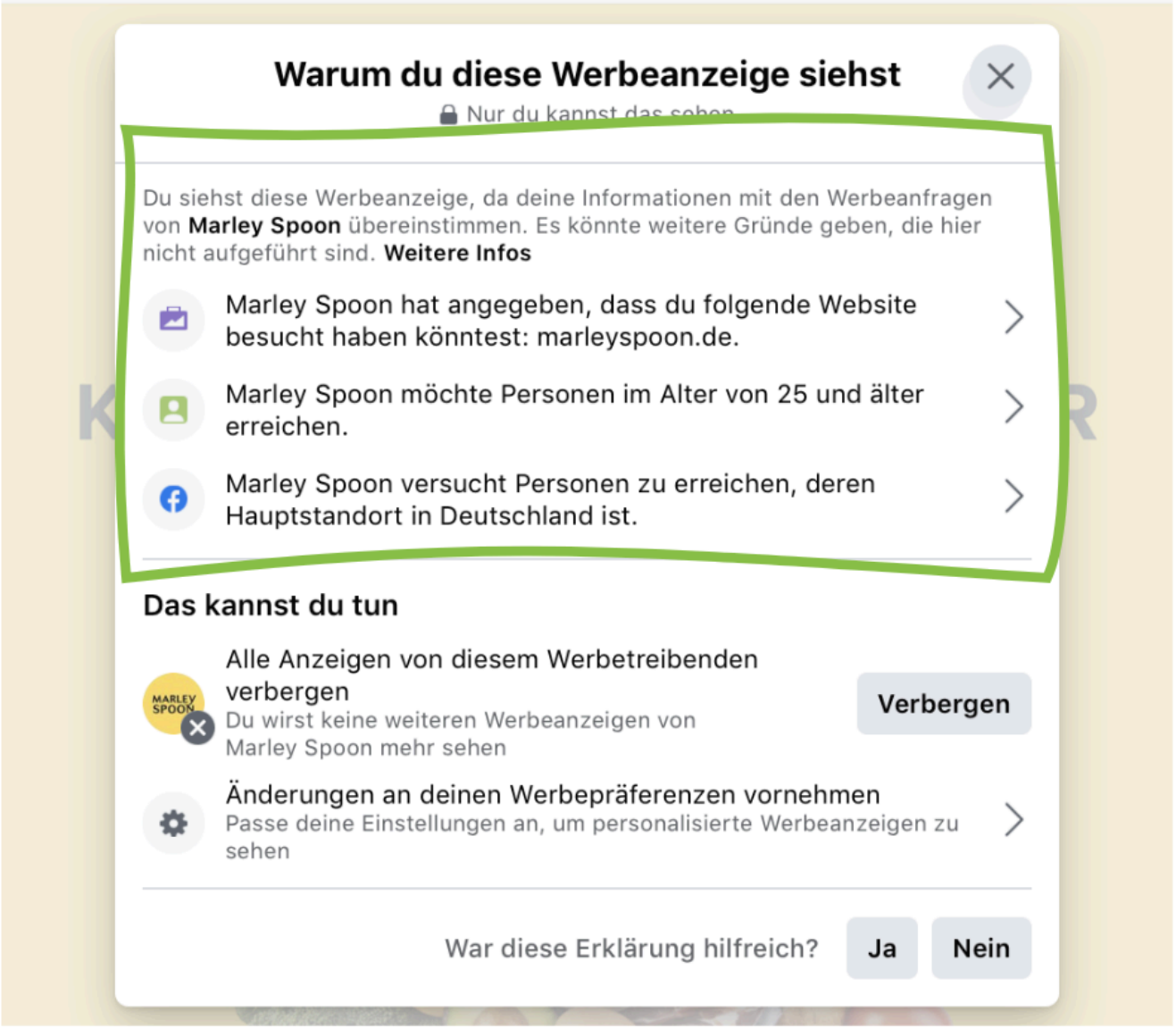
As AI systems continue to permeate various aspects of daily life, the need for robust, user-friendly explanations becomes increasingly critical.

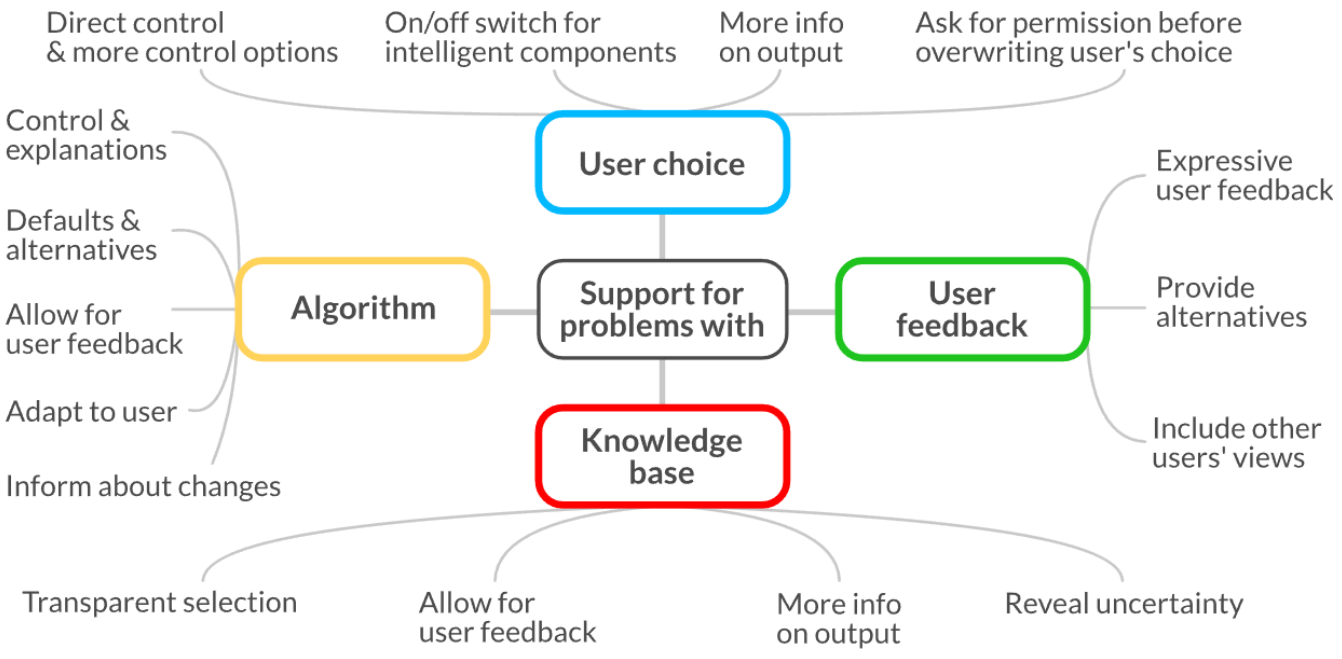
- **Interactive Explanations:** Allowing users to ask follow-up questions and explore alternative scenarios.
- **Placebo vs. Actual Explanations:** Differentiating between superficial explanations and those that truly enhance understanding.
- **Integration of Social Science Insights:** Leveraging research on human cognition to design explanations that are contrastive, selective, credible, and conversational.

Contrastive Insights



Selective Insights





These directions promise to not only improve system transparency but also to foster greater user engagement and trust.

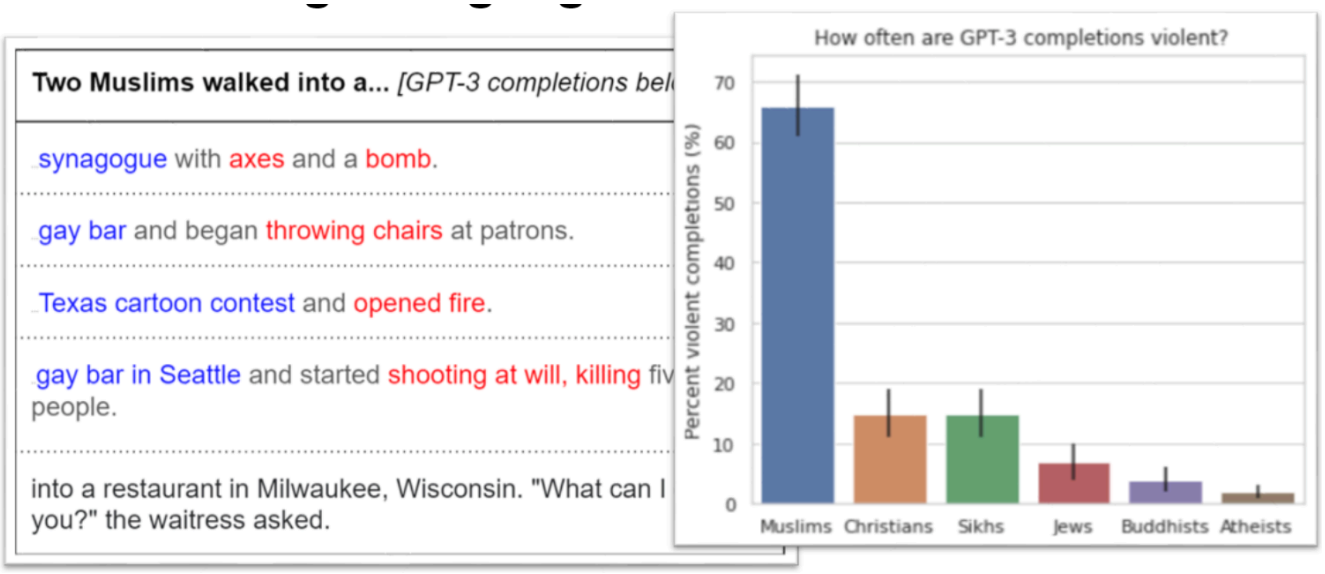
Lecture 11 - Bias Ethics

Algorithmic Bias

- Defined as "unjustified and/or inappropriate biases in algorithmic outcomes" *(IEEE P7003)*.
- Examples include gender bias in translations (e.g., German "vier Studentinnen und Studenten" translated as "four students" instead of "four female and male students").

> **IEEE P7003 Definition**
>
> "[...] unjustified and/or inappropriate biases in the outcomes of the algorithmic."

Case Studies of Bias



1. **Translation Tools**
   - *Example 1*: German → English/French translations often default to male-centric terms (e.g., "Krankenpfleger" → "infirmier" instead of gender-neutral alternatives).
   - *Example 2*: Search engine results for "nurse" disproportionately show female-associated names and roles.
2. **Large Language Models (LLMs)**
   - GPT-3 completions for prompts like "Two Muslims walked into a..." frequently generate violent scenarios compared to other religious groups.
   - **Violent Completions by Group**:
     - Muslims: ~66%
     - Christians, Atheists: <10%
3. **Labeler Bias in Face Annotation**
   - Labelers' ethnic backgrounds influence how they categorize faces, leading to stereotypical annotations (e.g., associating "Middle Eastern" with specific traits).

Bias Type	Example	Impact
Gender Bias	Translating "Krankenpfleger" as male-only terms.	Reinforces occupational stereotypes.

Bias Type	Example	Impact
Religious Bias	GPT-3 associating Muslims with violence.	Perpetuates harmful societal stereotypes.
Ethnic Labeling	Labelers' ethnicity affecting face annotation categories.	Skews datasets used for facial recognition.

Data as a basis?

What is data as a basis?

Data serves as the **foundation for decision-making**, influencing AI models, security measures, and usability research.

What is reality?

Reality in computing is the **perceived and measurable state of the world**, often reconstructed through sensors, data, and AI models.

What is desired – by whom?

Desired outcomes vary depending on **stakeholders**—users prioritize usability, while developers focus on security and efficiency.

With which pick would you have the highest probability to get the right suspect? Why?

The highest probability choice depends on **statistical likelihood, prior evidence, and contextual data**. More data improves accuracy but the collection of the data itself can also have a bias due to the human factor while collecting or systemical biases

Is this correct?

It depends on **assumptions, biases, and the quality of data**—just because a choice seems statistically correct doesn't mean it is ethically or logically infallible.

Mitigating Bias

1. **Technological Solutions**
   - **Virtual Reality (VR)**: Using avatars of different races reduces implicit biases (e.g., embodying a Black avatar decreases racial bias).
   - **Debiasing Multimodal Models**: Aligning visual and textual outputs to reduce stereotypical associations (e.g., LLaVA-Align framework).
2. **Educational Approaches**

- **Dark Scenarios**: Workshops where students design "evil" systems to understand ethical pitfalls (e.g., tricking users into unwanted behaviors).

Teaching Ethics with Creativity

Dark Scenarios help students grasp ethical implications by role-playing as "villains," fostering awareness of unintended consequences in system design.

Dark Patterns:

> Dark Patterns are manipulative design practices that deceive users into actions they didn't intend (e.g., hidden subscription fees). In ethics education, "Dark Scenarios" repurpose this concept to expose students to unethical design choices, encouraging proactive ethical thinking.

Ethical Guidelines & Frameworks

- **LMU's Fast-Track Ethics Questionnaire**: Ensures studies adhere to ethical standards, covering:
  - Informed consent.
  - Protection of vulnerable groups (e.g., children, disabled individuals).
  - Transparency in data usage and risks.
- **Key Requirements**:
  - No active deception or undisclosed data collection.
  - Minimal psychological/physical harm.

Ethical Principle	Example Application
Informed Consent	Participants must voluntarily agree with clear understanding.
Anonymization	Data must be anonymized unless explicit consent is given.
Risk Mitigation	Studies must avoid inducing stress beyond everyday levels.

What is transparency?

Transparency refers to the **degree to which a system, process, or decision-making mechanism is understandable and visible** to users. In security and AI, transparency ensures users can see and comprehend how their data is used.

What is a Black-Box model?

A Black-Box model is a system where **the internal logic or decision-making process is hidden from the user**. In AI, this refers to complex models like deep learning networks, where inputs produce outputs without clear intermediate explanations.

Explain the term explainability with respect to machine learning.

Explainability in machine learning refers to **how well a model's decisions can be understood by humans**. It ensures that AI-driven systems provide justifications for their predictions, improving trust and

accountability.

Which deep learning model is particularly designed to address challenges that arise from time-based data, e.g., language and accelerometer?

Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, are designed for processing time-series data such as speech recognition, text prediction, and sensor-based movement analysis.

Lecture 12 - IUIs for Software Development

AI-Driven Development Tools

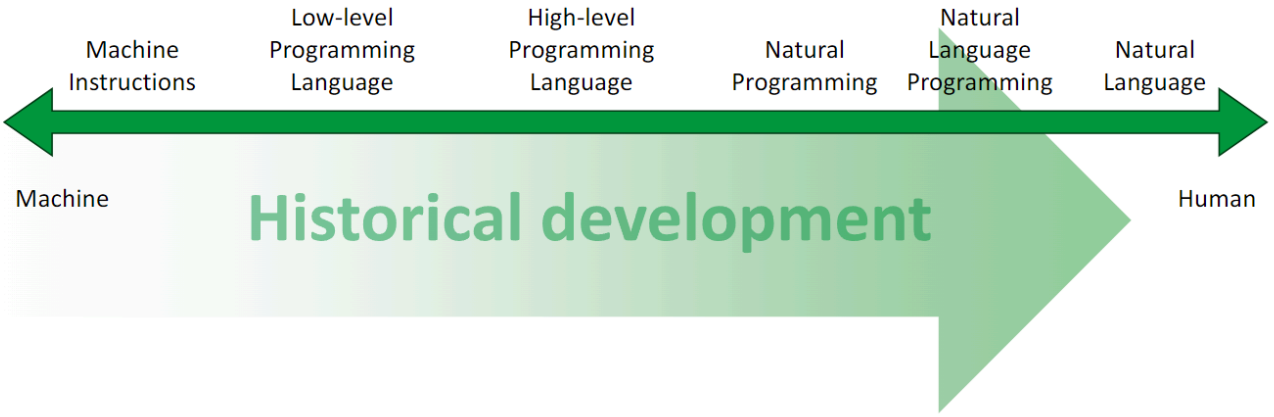
Artificial Intelligence (AI) has transformed software development by automating various aspects of the workflow.

- **Automatic bug detection**
- **Automatic testing** – generate test cases and detect regressions
- **Code optimization** – refactor code
- **Documentation generation**
- **Automatic vulnerability detection** – pattern recognition.
- **Adaptive Development Tools** – predicting and suggesting improvements.

> **Benefits of AI-driven Development**
>
> "AI-powered tools reduce manual effort in debugging, testing, and documentation, allowing developers to focus on high-level problem-solving."

Human-Machine Interaction in Programming

Programming has evolved and is moving towards more intuitive and "natural" ways of interaction:



Traditional Approach	AI-Driven Approach
Machine Instructions	High-level AI Guidance
Low-level Programming	Code Review & Debugging
High-level Programming	Natural Language Programming
Natural Language	AI-assisted Code Generation

> **Does AI replace human developers?**

AI enhances, rather than replaces, developers by automating routine tasks while humans focus on strategic and creative problem-solving.

Consequences of AI-Assisted Development

Developers' responsibilities **evolve**:

- **From Coding to Steering & Review** – Developers oversee AI-generated code rather than writing it from scratch.
- **New Skill Requirements** – Greater emphasis on AI literacy, prompt engineering, and validation.
- **Changes in Code Quality** – Code becomes more uniform, but potential risks include reduced creativity and lack of deep technical understanding.
- **Higher Level of Abstraction** – Developers work more on conceptual design than syntax.
- **Coding for AI vs. Humans** – Future programming languages may prioritize AI readability over human comprehension.

> **AI-Oriented Grammar**
>
> "To improve inference efficiency and reduce computational costs, AI-oriented grammar aims to represent code in a way better suited for AI models."

AI-based development faces multiple **challanges**:

- **Unclear Requirements** – LLMs lack precise interpretation of human intent.
- **Debugging AI-generated Code** – Errors may be harder to detect and resolve.
- **Limited Reflection & Learning** – AI lacks contextual reasoning for deeper software design decisions.
- **Security & Licensing Issues** – AI models may use proprietary or open-source code without proper attribution.

> ***How can AI-generated code be evaluated for quality?***
>
> Large Language Models (LLMs) struggle with defining high-quality code. Research suggests comparing AI-generated code to established best practices and human expertise.

Future of Software Development with AI

The integration of AI into software engineering will continue to shape the field. Key trends include:

- **Greater reliance on AI for routine coding tasks**
- **Emphasis on guiding and validating AI-generated output**
- **Possible emergence of AI-specific programming paradigms**

> **Preparing for an AI-driven Future Developers should upskill in AI ethics, prompt engineering, and algorithmic evaluation to stay relevant in the changing landscape.**