# https://elicit.com/notebook/f154a797-0602-43dd-8e3f-93c5f051bd3d

# From Concept to Contribution: A Guide to Developing a Publishable Undergraduate Research Project in Explainable Dark Pattern Detection

## Section 1: Identifying a Research-Worthy Project: The Case for Explainable Dark Pattern Detection

The selection of a research project, particularly at the undergraduate level, requires a delicate balance between ambition and feasibility. An ideal project occupies a niche where a tangible technical contribution can be made within a constrained timeframe, yet this contribution must also address a novel question within the existing academic landscape. This report outlines such a project at the intersection of Human-Computer Interaction (HCI), Applied Artificial Intelligence (AI), and digital ethics. The proposed work involves the design, implementation, and evaluation of a system for the automated detection and, crucially, the explanation of "dark patterns" in web interfaces. This area is not only technically rich but also addresses a socio-technical problem of increasing urgency, making it a fertile ground for a high-impact, publishable undergraduate research project.

### 1.1. Introduction to Dark Patterns as a Socio-Technical Problem

In the digital commons, user interfaces (UIs) serve as the primary medium of interaction between users and services. Ideally, these interfaces are designed to be intuitive, efficient, and empowering. However, a growing body of evidence points to the proliferation of manipulative design practices known as "dark patterns." A dark pattern is formally defined as a user interface that has been carefully crafted to trick users into doing things they did not mean to do, such as buying insurance with their purchase or signing up for recurring bills.1 These are not simple design errors but deliberate, malicious UI choices that exploit cognitive biases to lead users toward actions that benefit the service provider, often at the user's expense.2

The impact of dark patterns extends far beyond mere user frustration. They constitute a significant socio-technical issue with profound ethical, privacy, and financial ramifications. For instance, patterns categorized as "Forced Action" may compel a user to sign up for a newsletter to access content, while "Hidden Costs" can add unexpected charges at the final stage of a checkout process.5 These practices erode consumer trust, undermine user autonomy, and create a coercive digital environment where informed consent becomes illusory.2 The problem is so pervasive that it has garnered the attention of international regulatory bodies. The European Data Protection Board (EDPB), for example, has issued guidelines on identifying and avoiding dark patterns in social media interfaces, recognizing their potential to subvert data protection principles.8 Similarly, the Organisation for Economic Co-operation and Development (OECD) has highlighted the risks dark patterns pose to consumer behavior and privacy.8 This high-level regulatory concern underscores the topic's contemporary relevance and the pressing need for technical solutions that can empower and protect users.

### 1.2. The Novelty Frontier: Why Detection is Not Enough

In response to the proliferation of dark patterns, the academic and open-source communities have shifted from merely cataloging these deceptive designs to building systems for their automated detection. A review of current research and available tools reveals an active and evolving field. Numerous projects leverage machine learning, particularly Natural Language Processing (NLP), to identify text-based dark patterns on e-commerce sites and other platforms.4 Prototypes, often in the form of browser extensions, have been developed to scan webpage content and flag potential manipulations.1

Despite this progress, significant gaps remain. Existing detection tools often suffer from critical limitations, including incomplete coverage of the ever-growing taxonomy of dark patterns and challenges with detection accuracy.12 The dynamic nature of web design means that new, more subtle forms of manipulation are constantly emerging, creating a cat-and-mouse game between designers of deceptive interfaces and the tools built to detect them.

However, a more fundamental limitation of many current detection systems lies in their "black box" nature. These tools may successfully flag a piece of text or a UI element as a dark pattern—for example, by highlighting a countdown timer—but they often fail to provide the user with a clear, comprehensible rationale for this classification. This lack of transparency limits the tool's utility in two crucial ways. First, it hinders user trust; without understanding *why* a tool made a certain decision, a user may be skeptical of its accuracy or dismiss its warnings. Second, and more importantly from an HCI perspective, it represents a missed opportunity for education. A tool that simply blocks or flags a pattern treats the user as a passive subject to be protected. In contrast, a tool that explains *why* a design is manipulative can actively foster critical awareness, empowering the user to recognize and resist similar patterns in the future, even without the tool's assistance. The research frontier, therefore, is moving beyond the technical act of detection and toward the HCI challenge of making that detection meaningful and empowering for the end-user.

### 1.3. Introducing Explainable AI (XAI) as the Research Contribution

The "black box" problem is not unique to dark pattern detection; it is a central challenge in the broader field of artificial intelligence. In response, the subfield of Explainable AI (XAI) has emerged with the goal of developing techniques that produce more transparent and interpretable models, helping humans understand and trust their outputs.13 One of the most powerful and versatile XAI frameworks is SHAP (SHapley Additive exPlanations).14

SHAP is a game-theoretic approach that explains the prediction of any machine learning model by computing the contribution of each feature to that prediction.14 It is based on Shapley values, a concept from cooperative game theory that ensures a fair distribution of a "payout" (the model's prediction) among the "players" (the input features). In the context of text classification, SHAP can quantify precisely how much each word or token in a sentence contributed to the final classification. For example, it can determine that in the phrase "Hurry, only 2 items left in stock!", the words "Hurry" and "only 2 left" strongly pushed the model's prediction towards an "Urgency" classification, while other words had a negligible impact.

The application of XAI, and specifically SHAP, to the problem of dark pattern detection represents a clear and compelling research novelty. The contribution of this project is not simply to build another detector. Instead, the core innovation is to create an *interpretable* detector. By integrating a SHAP-based explanation module, the system can move beyond a binary "is/is not a dark pattern" output and provide users with nuanced, evidence-based insights into the manipulative nature of the language they encounter. This fusion of state-of-the-art NLP for detection with state-of-the-art XAI for explanation directly addresses the critical limitation of existing tools and opens a new avenue for research focused on user empowerment and digital literacy.

### 1.4. Project Thesis: Building a Browser Extension to Detect and Explain Dark Patterns in Real-Time

Based on the identified problem and the proposed novel approach, this report outlines a project with the following formal thesis:

*To design, implement, and evaluate a browser extension that leverages a fine-tuned transformer model to detect text-based dark patterns on e-commerce websites and utilizes SHAP to provide real-time, human-readable explanations for its classifications, thereby enhancing user awareness and critical thinking.*

This thesis statement encapsulates a project that is unique, simple, and doable within a three-month undergraduate timeframe. It is unique because it combines dark pattern detection with XAI, a largely unexplored intersection. It is simple because it leverages existing, high-level libraries (Hugging Face Transformers, SHAP, Flask) and pre-existing datasets, avoiding the need to build complex systems from scratch. It is doable because it has a narrow, well-defined scope: text-based patterns on e-commerce sites, delivered via a browser extension.

Furthermore, this thesis naturally gives rise to a clear, evaluable research question: "Does providing SHAP-based explanations for dark pattern detections improve a user's understanding of and ability to recognize manipulative designs?" The choice to implement the system as a browser extension is a critical methodological decision. A dark pattern is an in-situ phenomenon, experienced within the context of a live, interactive webpage.2 A standalone application or a controlled lab study can only approximate this experience. A browser extension, however, allows the tool to be deployed and evaluated in the user's natural browsing environment.16 This enables a more ecologically valid user study, which is a cornerstone of high-quality HCI research and significantly strengthens the potential of the resulting research paper.

## Section 2: Architectural Blueprint of the System: "Ethical Eye"

To realize the project thesis, a robust and modular system architecture is required. This section provides a detailed technical blueprint for the proposed system, named "Ethical Eye".5 The architecture is divided into four primary components: a data-gathering content script, a dark pattern classification engine, an explanation module, and a user interface for visualization and communication. This design intentionally separates the lightweight, browser-based frontend from the computationally intensive machine learning backend, a critical pattern for ensuring both performance and feasibility within the project's constraints.

### 2.1. Component 1: The Data-Gathering Content Script

The first component of Ethical Eye is a content script, a JavaScript file that runs in the context of a webpage.18 Its sole responsibility is to act as the system's "eyes," extracting relevant textual content from the page's Document Object Model (DOM) for analysis.

* **Functionality:** The script, content.js, will be programmatically injected by the browser into target webpages. Upon page load, it will traverse the DOM and extract text from elements that commonly contain persuasive or descriptive language.
* **Implementation Details:** The script will utilize standard web APIs for DOM manipulation. Methods such as document.querySelectorAll will be used to select a collection of HTML elements based on their tags (e.g., p, span, div, h1, h2, h3, button). A loop will then iterate through these elements, extracting their textual content via the element.textContent property.18 This process will compile a list of text snippets from the page that will be sent to the backend for analysis.
* **Configuration:** The extension's manifest.json file is the central configuration hub. It will be configured to inject content.js automatically. The "content\_scripts" key will define the script to be injected, and the "matches" key will specify the URL patterns where the script should run, such as "\*://\*.amazon.com/\*" or "\*://\*.ebay.com/\*", ensuring the extension is active on relevant e-commerce sites.17

### 2.2. Component 2: The Dark Pattern Classification Engine

The core of the system's intelligence resides in the classification engine, a machine learning model trained to recognize manipulative language. This component will be implemented in Python and hosted on a local server.

* **Model Choice:** The recommended model is **DistilBERT** (distilbert-base-uncased). As a distilled version of the powerful BERT model, it is significantly smaller and faster while retaining approximately 97% of BERT's language understanding capabilities.21 This trade-off between performance and computational cost makes it an ideal choice for a project where inference speed for real-time feedback is important and where computational resources may be limited.22
* **Framework:** The Hugging Face transformers library will be the primary tool for this component. This library provides a high-level API that simplifies the process of working with state-of-the-art transformer models. Specifically, the AutoTokenizer class will be used to load the appropriate tokenizer for DistilBERT, which converts raw text into a numerical format the model can understand. The AutoModelForSequenceClassification class will be used to load the pre-trained DistilBERT model and automatically add a classification head (a simple neural network layer) on top, which will be fine-tuned for the dark pattern detection task.22
* **Dataset and Training:** The feasibility of this project within a three-month timeline hinges on leveraging existing resources rather than creating them from scratch. Training a large language model is computationally prohibitive for an undergraduate project.21 Therefore, the approach will be to fine-tune the pre-trained DistilBERT model. The training will use a publicly available, text-based dark pattern dataset, such as the one curated by Mathur et al. 9, which has been used as a benchmark in several subsequent research papers.8 The training process itself will be managed using the Hugging Face  
  Trainer API, which automates the training loop, evaluation, and model saving, allowing the developer to focus on high-level configuration rather than low-level implementation details.24 The model will be trained to classify text into several categories of dark patterns, as detailed in Table 1.

|  |  |  |
| --- | --- | --- |
| **Pattern Name** | **Definition** | **Example Text** |
| **Urgency** | Creating a sense of time pressure to rush a user's decision. | "Hurry, this offer expires in 05:32." |
| **Scarcity** | Implying that a product is in limited supply to increase its perceived value. | "Only 2 left in stock! Order soon." |
| **Social Proof** | Using the behavior of other users to influence a decision. | "23 people bought this in the last hour." |
| **Misdirection** | Using visual or linguistic cues to steer a user towards a particular choice. | "No thanks, I prefer to pay full price." (as the opt-out for a discount) |
| **Forced Action** | Requiring the user to perform an action to continue with their primary task. | "Sign up for our newsletter to read the rest of this article." |
| **Hidden Costs** | Revealing previously undisclosed charges late in the checkout process. | "Service fee: $4.99" (appears on the final confirmation screen) |
| Table 1: Taxonomy of Detectable Text-Based Dark Patterns. This table consolidates common patterns from academic literature 2 into a concrete set of target classes for the classification model, providing a clear and literature-grounded foundation for the machine learning task. |  |  |  |

### 2.3. Component 3: The Explanation Module (XAI Core)

This component is the central research contribution of the project. It takes the output of the classification engine and generates a human-readable explanation for its decision.

* **Framework:** The shap Python library will be used to implement this module. The library provides a shap.Explainer object that can be wrapped around a wide variety of models, including those from the Hugging Face transformers library. The integration is particularly seamless when using the pipeline abstraction from transformers, which bundles the tokenizer and model together.14
* **Integration:** The workflow is sequential. First, the classification engine (Component 2) receives a text snippet from the content script and makes a prediction (e.g., "Scarcity"). If a dark pattern is detected, that text snippet and the trained model are passed to the SHAP explainer. The explainer then computes the SHAP values for each token in the input text, quantifying its contribution to the "Scarcity" prediction.
* **Output:** The raw output from this module is a data structure (e.g., a list of tuples or a dictionary) that maps each word in the input text to its corresponding SHAP value. Words with high positive SHAP values are the primary drivers of the classification. This structured data is the raw material that will be sent back to the browser extension to be visualized for the user.

### 2.4. Component 4: The User Interface and Communication

The final component is responsible for tying the system together: communicating between the browser and the backend server, and presenting the results to the user in an intuitive way.

* **Communication Mechanism:** The architecture relies on a client-server model. The browser extension (client) cannot directly run the Python-based ML and XAI components due to the limitations of the browser environment; attempts to run complex Python libraries in-browser are notoriously unreliable and limited.28 Therefore, the  
  content.js script will use the standard fetch API to send an HTTP POST request to the backend server. The body of this request will contain the text snippets scraped from the webpage. The backend will process the text and send back a JSON response.
* **Flask Backend:** A lightweight Python web framework, Flask, will be used to create the backend server.31 The server will have a single API endpoint (e.g.,  
  /analyze). This endpoint will load the fine-tuned DistilBERT model and the SHAP explainer into memory upon startup. When it receives a request, it will pass the incoming text to the model for prediction and then to the explainer for SHAP value generation. Finally, it will serialize the prediction, the confidence score, and the SHAP values into a JSON object and return it as the HTTP response.33
* **Visualization:** Upon receiving the JSON response from the Flask server, the content.js script will parse the data and dynamically manipulate the DOM to present the findings to the user. This involves two actions:
  1. **Highlighting:** The script will locate the original text on the webpage that was flagged as a dark pattern and wrap it in a <span> element with a specific CSS class to visually highlight it (e.g., with a colored underline or background).
  2. **Explanation Tooltip:** An event listener will be attached to the highlighted <span>. When the user hovers their mouse over the highlighted text, a tooltip (implemented using a library like Tooltip.js or a custom div element) will appear. This tooltip will display the explanation, which is constructed from the SHAP values. For instance, it might read: "This text was flagged as **Urgency** with 98% confidence. The model focused on the words **'Hurry'** and **'expires'**." This provides an immediate, context-aware, and evidence-based explanation directly within the user's browsing flow, inspired by the visualization principles of shap.plots.text.14

## Section 3: A Three-Month Implementation and Evaluation Plan

A successful research project requires not only a sound technical architecture but also a disciplined and pragmatic project plan. Given the three-month constraint, the following schedule is designed to balance software development with the parallel research activities required to produce a publishable paper. The plan is divided into three month-long phases: Model Development, System Integration, and Evaluation/Writing. This structure allows for focused, iterative progress and de-risks the project by tackling the core machine learning challenges first.

### 3.1. Month 1: Model Development and Baseline Validation (Weeks 1-4)

The first month is dedicated to building and validating the Python-based backend components. This work can be conducted entirely within a development environment like a Jupyter or Google Colab notebook, independent of the browser extension frontend.

* **Weeks 1-2: Data Acquisition and Model Fine-Tuning**
  + **Tasks:** The initial priority is to establish the core classification capability. This involves acquiring a suitable dataset, such as the text-based dark pattern corpus from Mathur et al.'s 2019 study 9, and preprocessing it for training. A Python environment must be set up with the necessary libraries, including  
    transformers, datasets, torch, and scikit-learn. A training script will be written using the Hugging Face Trainer API to fine-tune a distilbert-base-uncased model on the preprocessed data.
  + **Deliverable:** A fine-tuned sequence classification model saved to disk, along with a script that can load the model and perform inference on new text samples. A preliminary performance report on a validation set should be generated. This phase follows established tutorials for text classification with Hugging Face.22
* **Weeks 3-4: Explainability Integration and API Development**
  + **Tasks:** With a working model, the focus shifts to integrating the XAI component. The shap library will be used to create an explainer object for the fine-tuned model. The task is to write a function that takes a text input, gets a prediction from the model, and then generates corresponding SHAP values that attribute the prediction to individual tokens.14 Subsequently, a simple Flask web server (  
    app.py) will be developed. This server will load the model and the SHAP explainer at startup and expose a single API endpoint. This endpoint will accept a JSON request containing text and return a JSON response with the classification label, confidence score, and the SHAP values.
  + **Deliverable:** A running Flask application that can be tested locally using a tool like curl or Postman. The API contract (request/response format) should be finalized and documented. This deliverable serves as the complete, testable backend for the browser extension.32

### 3.2. Month 2: System Integration and Prototyping (Weeks 5-8)

The second month focuses on building the frontend browser extension and integrating it with the now-complete backend.

* **Weeks 5-6: Browser Extension Frontend Development**
  + **Tasks:** This phase involves creating the foundational structure of the Chrome extension. This includes the manifest.json file, which defines the extension's permissions and capabilities, and the primary content.js script.17 The core logic for DOM traversal and text extraction will be implemented in  
    content.js. This script must efficiently scrape text from a webpage without significantly impacting performance. The fetch API call from content.js to the local Flask server will also be implemented and tested to ensure communication between the browser and the Python backend is functional.
  + **Deliverable:** A basic, loadable Chrome extension that, upon visiting a target webpage, successfully scrapes text content and sends it to the running Flask backend. The response from the backend should be visible in the browser's developer console.
* **Weeks 7-8: End-to-End Integration and UI Refinement**
  + **Tasks:** This is the critical integration phase where the full system comes together. The content.js script will be updated to parse the JSON response from the backend. Based on this data, it will dynamically modify the webpage's DOM to highlight the flagged text and generate the explanation tooltips. Significant effort during this phase should be dedicated to the HCI aspect of the design: the highlighting should be noticeable but not disruptive, and the tooltips should be clear, concise, and easy to understand. The goal is to create a seamless and informative user experience.35
  + **Deliverable:** A fully functional prototype of the "Ethical Eye" browser extension. The end-to-end data pipeline (scrape -> send to backend -> receive analysis -> display on page) should be working reliably on a set of test websites.

### 3.3. Month 3: Rigorous Evaluation and Paper Writing (Weeks 9-12)

The final month is dedicated to rigorously evaluating the system and translating the project's work into a formal research paper.

* **Week 9: Quantitative Performance Analysis**
  + **Tasks:** A formal, quantitative evaluation of the fine-tuned DistilBERT model is necessary for the research paper. Using a held-out portion of the dataset (a test set), the model's performance will be measured using standard text classification metrics: Accuracy, Precision, Recall, and F1-score.37 These metrics provide an objective measure of the detector's technical correctness.
  + **Deliverable:** A table of classification results and a confusion matrix. This will form the core of the "Quantitative Model Performance" subsection in the paper's evaluation section.5
* **Weeks 10-11: Qualitative User Study**
  + **Tasks:** The project's central thesis is about enhancing user awareness, a claim that cannot be substantiated by F1-scores alone. A qualitative user study is therefore methodologically essential.36 A small-scale study should be designed, recruiting 5-10 participants. Participants will be asked to perform a series of tasks (e.g., "find and purchase a specific item") on several e-commerce websites while using the Ethical Eye extension. The "think-aloud" protocol will be used, where participants verbalize their thoughts as they interact with the sites and the extension. Post-task, semi-structured interviews will be conducted to gather deeper insights into the usability of the extension and the perceived value of the XAI-driven explanations.
  + **Deliverable:** A set of interview transcripts and observation notes. Thematic analysis will be performed on this data to identify recurring patterns, user opinions, and key insights. These findings, supported by illustrative quotes, will form the "Qualitative User Study Findings" subsection of the paper.
* **Week 12: Finalize Paper and Prepare for Submission**
  + **Tasks:** With all technical work and evaluation complete, the final week is dedicated to writing. The qualitative data from the user study will be analyzed to draw conclusions about the effectiveness of the explainable approach. The Discussion and Conclusion sections of the paper will be written, interpreting the combined quantitative and qualitative results. The entire manuscript will be polished, ensuring it follows the structure outlined in Section 4.
  + **Deliverable:** A complete, submission-ready research paper.

This dual-evaluation strategy, combining hard quantitative metrics with rich qualitative user feedback, is a hallmark of strong HCI research. A purely technical paper would be limited to reporting the model's accuracy. By incorporating a user study, the project can address its core HCI-focused thesis. This mixed-methods approach allows the resulting paper to tell a far more compelling narrative: not only was a tool built that is technically accurate (the quantitative results), but it is also a tool that is genuinely useful and empowering to its users (the qualitative results). This significantly elevates the project's potential for publication in a reputable academic venue.

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Primary Task** | **Deliverable** | **Relevant Sources** |
| **1-2** | Data Acquisition & Model Fine-Tuning | Fine-tuned DistilBERT model saved to disk; validation accuracy report. | 9 |
| **3-4** | XAI Integration & Flask API Development | Running Flask server with a single API endpoint for classification and explanation. | 14 |
| **5-6** | Browser Extension Frontend Development | Basic Chrome extension that scrapes text and communicates with the Flask backend. | 17 |
| **7-8** | End-to-End Integration & UI Refinement | Fully functional "Ethical Eye" prototype with highlighting and explanation tooltips. | 19 |
| **9** | Quantitative Performance Analysis | Table of classification metrics (Accuracy, Precision, Recall, F1-score) on the test set. | 5 |
| **10-11** | Qualitative User Study | User study protocol, interview transcripts, and thematic analysis of findings. | 36 |
| **12** | Finalize Research Paper | Complete, polished, submission-ready manuscript. | 40 |
| *Table 2: 12-Week Detailed Project Schedule. This schedule provides an actionable timeline, breaking the project into manageable weekly goals and linking tasks to relevant source materials to guide implementation.* |  |  |  |

## Section 4: Structuring and Writing Your Research Paper

The culmination of a research project is the formal communication of its methods, findings, and contributions to the academic community. A well-structured paper is essential for conveying the work's value and rigor. This section provides a template for the research paper resulting from the "Ethical Eye" project, along with strategic advice on framing the contribution and selecting an appropriate publication venue.

### 4.1. A Section-by-Section Paper Template

Adhering to a standard structure will ensure the paper is clear, logical, and meets the expectations of academic reviewers. The following template is standard for papers in computer science and HCI.

* **Abstract:** A dense, single-paragraph summary (typically 150-250 words) of the entire paper. It should concisely state the problem (manipulative dark patterns), the gap in existing work (lack of explainability), the proposed method (a browser extension using a transformer model and SHAP), the key results (e.g., high classification accuracy and positive user feedback on explanations), and the primary contribution (demonstrating the value of XAI for user empowerment).
* **Introduction:** This section sets the stage. It should begin by motivating the importance of the dark patterns problem, citing their negative impact on users. It should then briefly survey the state of the art in detection, leading to the identification of the research gap: current tools are often "black boxes." This is where the paper introduces its core idea—the integration of XAI. The introduction must conclude with a clear thesis statement and a list of the paper's specific contributions (e.g., 1. The design and implementation of Ethical Eye, a novel XAI-driven detection tool; 2. A quantitative evaluation of the tool's classification accuracy; 3. A qualitative user study assessing the impact of explanations on user awareness).
* **Related Work:** This section demonstrates a thorough understanding of the existing literature and situates the project within the broader academic conversation. It should be organized thematically into at least two subsections:
  + *Dark Pattern Taxonomies and Automated Detection:* Review key papers that define and categorize dark patterns, as well as prior work on building automated detectors.
  + *Explainable AI for Natural Language Processing:* Discuss the rise of XAI and review applications of techniques like LIME and SHAP, particularly for explaining text classification models.
* **System Design:** This is the primary technical section. It should provide a detailed description of the "Ethical Eye" architecture, as outlined in Section 2 of this report. Describe each of the four components: the content script, the classification engine (including model choice and training details), the XAI module, and the UI/communication layer. An architectural diagram illustrating the data flow from the browser to the Flask server and back is highly recommended.
* **Evaluation:** This section presents the results of the project's evaluation, corresponding to the plan in Section 3. It should be divided into two clear subsections:
  + *Quantitative Model Performance:* Present the results of the classification model on the held-out test set. Include the table of metrics (Accuracy, Precision, Recall, F1-score) and briefly interpret the numbers.
  + *Qualitative User Study Findings:* Describe the user study methodology (participants, tasks, procedure). Present the findings from the thematic analysis of the think-aloud and interview data. Use direct, anonymized quotes from participants to support the claims and bring the user experience to life.
* **Discussion:** This is where the results are interpreted and their implications are explored. What do the combined quantitative and qualitative findings mean? Discuss the broader implications of the work, such as the potential for XAI to be a standard feature in consumer protection tools. It is also crucial to honestly acknowledge the limitations of the study (e.g., the small number of participants in the user study, the focus on only text-based patterns, the use of a single language). Finally, suggest concrete directions for future research (e.g., extending the detector to handle visual dark patterns, testing on a larger and more diverse user population).
* **Conclusion:** Briefly reiterate the paper's main points: the problem, the method, and the key takeaway. It should leave the reader with a clear understanding of what was accomplished and why it matters.

### 4.2. Framing Your Contribution

The framing of the project's contribution is critical. A common mistake in undergraduate research is to frame the contribution too narrowly around the technical artifact itself. The contribution should be positioned not as "I built a dark pattern detector," but rather as the new knowledge generated by the project.

A stronger framing would be: *"This work presents the design and evaluation of an XAI-driven dark pattern detection system. Through a mixed-methods evaluation, we demonstrate that our system not only achieves high classification accuracy but that its SHAP-based explanations significantly enhance user understanding of manipulative designs. Our findings suggest that explainability is a key feature for the next generation of user empowerment tools."*

This framing elevates the project from a simple software engineering exercise to a piece of HCI research that generates new insights about the relationship between users, AI systems, and manipulative interfaces.

### 4.3. Curated List of Suitable Publication Venues

Selecting the right venue for submission is a strategic decision. The goal is to find a venue where the work will be appreciated by a relevant audience and has a realistic chance of acceptance. For an undergraduate project of this nature, a tiered approach is recommended.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Venue Name** | **Type** | **Focus Area** | **Prestige/Acceptance Rate** | **Submission Cycle** |
| Southern California Conference on Undergraduate Research (SCCUR) 41 | Undergraduate Conference | General UG Research | High acceptance rate; focused on presentation experience. | Typically Fall |
| Posters on the Hill (POH) 41 | Undergraduate Poster Session | General UG Research, Policy Focus | Highly competitive; prestigious for UG research. | Typically Spring |
| CHI Student Research Competition (SRC) 40 | Workshop/Competition at Main Conference | Human-Computer Interaction (HCI) | Competitive; excellent for networking and feedback from top HCI researchers. | Typically Fall/Winter |
| ICSE Student Research Competition (SRC) 40 | Workshop/Competition at Main Conference | Software Engineering | Competitive; suitable if the paper emphasizes the system's architecture and engineering. | Typically Fall |
| WACV Workshops 42 | Workshop at Main Conference | Computer Vision, HCI | Varies by workshop; good for applied AI projects with a user-facing component. | Typically Fall |
| *Table 3: Comparison of Potential Academic Venues. This table provides a strategic roadmap for disseminating the research, allowing for the selection of an appropriate venue based on the project's final quality and the student's career objectives.* |  |  |  |  |

The most suitable initial targets are undergraduate-focused conferences like SCCUR or student research competitions at major conferences like CHI or ICSE. These venues are specifically designed to showcase student work and provide a supportive environment for first-time presenters. They offer an excellent opportunity to receive feedback from senior researchers and to begin building an academic network.

## Section 5: An Alternative Pathway: Sentiment Analysis for Low-Resource Languages

While the primary proposal focuses on explainable dark pattern detection, an alternative project with a similar scope and potential for a research contribution lies in the domain of Natural Language Processing for low-resource languages. This alternative is presented to offer a choice that may align differently with specific interests or available datasets.

### 5.1. Project Concept: A High-Performance Sentiment Classifier for Marathi News

The vast majority of advances in NLP have been concentrated on high-resource languages like English, which benefit from massive, readily available datasets. This creates a significant technology gap for the billions of people who communicate in low-resource languages. A compelling undergraduate research project can be built around addressing this gap for a specific language, such as Marathi, which is spoken by over 83 million people yet has relatively few public-domain NLP resources.43

**Project Goal:** To build and rigorously evaluate an end-to-end system for sentiment analysis of Marathi news articles. The project's core would be a comparative study of different methods for creating a labeled dataset, which is the primary bottleneck for low-resource languages.

### 5.2. Implementation and Research Angle

This project would also be feasible within a three-month timeframe and would produce both a functional software system and a valuable research contribution.

* **Data Collection:** The first step would be to create a corpus of raw text. This would involve using Python libraries like requests and BeautifulSoup to scrape the headlines and body text of articles from prominent Marathi news websites, such as Lokmat, Saamana, and Maharashtra Times.46 This process would yield a large, domain-specific, unlabeled dataset.
* **Dataset Creation (The Research Core):** The central research activity would be to create labeled datasets for training a sentiment classifier using two distinct, competing methodologies. This comparative approach is the key to the project's novelty.
  1. **Machine Translation Approach:** Following the methodology proposed in recent literature 49, a large, existing English sentiment dataset (e.g., the Sentiment140 dataset containing 1.6 million labeled tweets) would be translated into Marathi using a high-quality machine translation API (e.g., Google Translate API). This creates a large but potentially noisy, out-of-domain dataset.
  2. **Lexicon-Based Approach:** Drawing on methods discussed for other Indian languages 45, a sentiment lexicon for Marathi would be curated. This involves creating a list of positive words (e.g., "उत्तम" - excellent, "चांगले" - good) and negative words (e.g., "वाईट" - bad, "खराब" - poor). This lexicon would then be used to programmatically assign sentiment labels (positive, negative, neutral) to the scraped Marathi news articles based on the prevalence of words from the lexicon. This creates a smaller, potentially more biased, but domain-specific dataset.
* **Modeling and Evaluation:** A multilingual transformer model, such as mBERT (multilingual BERT) or XLM-RoBERTa, would be fine-tuned for sentiment classification.50 Crucially, two separate models would be trained: one on the machine-translated dataset and one on the lexicon-labeled dataset. Both models would then be evaluated on a common, manually annotated test set of Marathi news articles to ensure a fair comparison.
* **Research Paper and Contribution:** The resulting research paper would present a rigorous comparative analysis of the two models. It would directly address the research question: *"For low-resource sentiment analysis in Marathi, is it more effective to fine-tune a model on a large, machine-translated, out-of-domain dataset or a smaller, lexicon-labeled, domain-specific dataset?"* Answering this question provides a valuable, practical contribution to the field of low-resource NLP. A significant secondary contribution would be the public release of the newly created and annotated Marathi news sentiment dataset, providing a valuable resource for future researchers.

This alternative project, like the primary proposal, is simple, unique, and doable. It leverages existing tools in a novel comparative study, leading to a clear and publishable research outcome.

#### Works cited

1. Carmineh/Dark-Pattern-Identifier: Browser Extension that ... - GitHub, accessed on August 10, 2025, <https://github.com/Carmineh/Dark-Pattern-Identifier>
2. Full article: Detecting dark patterns in shopping websites – a multi-faceted approach using Bidirectional Encoder Representations From Transformers (BERT) - Taylor & Francis Online, accessed on August 10, 2025, <https://www.tandfonline.com/doi/full/10.1080/17517575.2025.2457961>
3. yamanalab/why-darkpattern: [Proc of IEEE BigData 2023 ... - GitHub, accessed on August 10, 2025, <https://github.com/yamanalab/why-darkpattern>
4. Unmasking Dark Patterns: A Machine Learning Approach to Detecting Deceptive Design in E-commerce Websites - arXiv, accessed on August 10, 2025, <https://arxiv.org/html/2406.01608v1>
5. (PDF) Ethical Eye: Dark Pattern Detection on Websites - ResearchGate, accessed on August 10, 2025, <https://www.researchgate.net/publication/387506054_Ethical_Eye_Dark_Pattern_Detection_on_Websites>
6. Automated Detection of Dark Patterns in Website Design: Enhancing User Trust and Online Transparency, accessed on August 10, 2025, <https://norma.ncirl.ie/8232/1/sundarayyappanmuthukumarasamy.pdf>
7. Dark Patterns in AI-Enabled Consumer Experiences, accessed on August 10, 2025, <https://casmi.northwestern.edu/research/projects/dark-patterns.html>
8. Dark patterns in e-commerce: a dataset and its baseline evaluations - arXiv, accessed on August 10, 2025, <https://arxiv.org/html/2211.06543v2>
9. yamanalab/ec-darkpattern: [IEEE BigData 2022] Dark patterns in e-commerce: a dataset and its baseline evaluations - GitHub, accessed on August 10, 2025, <https://github.com/yamanalab/ec-darkpattern>
10. This repository contains the replication package of our ICSE'23 paper "AidUI: Toward Automated Recognition of Dark Patterns in User Interfaces" - GitHub, accessed on August 10, 2025, <https://github.com/SageSELab/AidUI>
11. Dark Pattern Detection Project - Dapde, accessed on August 10, 2025, <https://dapde.de/en/>
12. A Comprehensive Study on Dark Patterns - arXiv, accessed on August 10, 2025, <https://arxiv.org/html/2412.09147v1>
13. Explain text classification model predictions using Amazon SageMaker Clarify - AWS, accessed on August 10, 2025, <https://aws.amazon.com/blogs/machine-learning/explain-text-classification-model-predictions-using-amazon-sagemaker-clarify/>
14. shap/shap: A game theoretic approach to explain the output of any machine learning model. - GitHub, accessed on August 10, 2025, <https://github.com/shap/shap>
15. Learn Explainable AI: Introduction to SHAP Cheatsheet - Codecademy, accessed on August 10, 2025, <https://www.codecademy.com/learn/learn-explainable-ai/modules/introduction-to-shap/cheatsheet>
16. Extensions / Get started - Chrome for Developers, accessed on August 10, 2025, <https://developer.chrome.com/docs/extensions/get-started>
17. Let's Build a Browser Extension! - Digital Initiatives at the Grad Center, accessed on August 10, 2025, <https://gcdi.commons.gc.cuny.edu/2025/03/28/lets-build-a-browser-extension/>
18. Content scripts | Chrome Extensions, accessed on August 10, 2025, <https://developer.chrome.com/docs/extensions/develop/concepts/content-scripts>
19. Let's create a simple chrome extension to interact with DOM | by Nithyanandam Venu, accessed on August 10, 2025, <https://medium.com/@divakarvenu/lets-create-a-simple-chrome-extension-to-interact-with-dom-7bed17a16f42>
20. Run scripts on every page | Chrome Extensions, accessed on August 10, 2025, <https://developer.chrome.com/docs/extensions/get-started/tutorial/scripts-on-every-tab>
21. Fine-tune BERT Model for Sentiment Analysis in Google Colab - Analytics Vidhya, accessed on August 10, 2025, <https://www.analyticsvidhya.com/blog/2021/12/fine-tune-bert-model-for-sentiment-analysis-in-google-colab/>
22. Text classification - Hugging Face, accessed on August 10, 2025, <https://huggingface.co/docs/transformers/tasks/sequence_classification>
23. Text classification - Hugging Face, accessed on August 10, 2025, <https://huggingface.co/docs/transformers/v4.40.1/tasks/sequence_classification>
24. Fine-Tuning BERT for Sentiment Analysis: A Practical Guide | by Hey Amit - Medium, accessed on August 10, 2025, <https://medium.com/@heyamit10/fine-tuning-bert-for-sentiment-analysis-a-practical-guide-f3d9c9cac236>
25. Text Classification with Hugging Face Transformers Tutorial, accessed on August 10, 2025, <https://www.learnhuggingface.com/notebooks/hugging_face_text_classification_tutorial>
26. Build a Text Classifier with Transformers in 5 minutes - YouTube, accessed on August 10, 2025, <https://www.youtube.com/watch?v=8yrD0hR8OY8>
27. shap/shap: A game theoretic approach to explain the output ... - GitHub, accessed on August 10, 2025, <https://github.com/slundberg/shap>
28. Creating a Browser Extension using Python, accessed on August 10, 2025, <https://python-forum.io/thread-41455.html>
29. Creating a Browser Extension using Python - The freeCodeCamp Forum, accessed on August 10, 2025, <https://forum.freecodecamp.org/t/creating-a-browser-extension-using-python/665366>
30. Is it possible to create a Google chrome extension with Python? : r/learnpython - Reddit, accessed on August 10, 2025, <https://www.reddit.com/r/learnpython/comments/11tbste/is_it_possible_to_create_a_google_chrome/>
31. Twitter Sentiment Analysis WebApp using Flask - GeeksforGeeks, accessed on August 10, 2025, <https://www.geeksforgeeks.org/python/twitter-sentiment-analysis-webapp-using-flask/>
32. edwinrlambert/Sentiment-Analysis-Using-Flask - GitHub, accessed on August 10, 2025, <https://github.com/edwinrlambert/Sentiment-Analysis-Using-Flask>
33. Tutorial - Deploy an app for sentiment analysis with Hugging Face and Flask - OVHcloud, accessed on August 10, 2025, <https://help.ovhcloud.com/csm/en-public-cloud-ai-deploy-flask-hugging-face-sentiment-analysis?id=kb_article_view&sysparm_article=KB0048106>
34. Create A Chrome Extension in HTML CSS & JavaScript - GeeksforGeeks, accessed on August 10, 2025, <https://www.geeksforgeeks.org/javascript/create-a-chrome-extension-in-html-css-javascript/>
35. Illuminating the Dark: Designing for a Dark Pattern Detection Tool - Bergen Open Research Archive, accessed on August 10, 2025, <https://bora.uib.no/bora-xmlui/bitstream/handle/11250/3206844/64325789.pdf?sequence=1>
36. Watching Them Watching Me: Browser Extensions' Impact on User Privacy Awareness and Concern, accessed on August 10, 2025, <https://www.ndss-symposium.org/wp-content/uploads/2017/09/watching-them-watching-me-browser-extensions-impact-on-user-privacy-awareness-and-concerns.pdf>
37. A review of sentiment analysis: tasks, applications, and deep learning techniques, accessed on August 10, 2025, <https://www.researchgate.net/publication/381881700_A_review_of_sentiment_analysis_tasks_applications_and_deep_learning_techniques>
38. A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research - MDPI, accessed on August 10, 2025, <https://www.mdpi.com/2076-3417/13/7/4550>
39. Usability Evaluation of User Interfaces - WebTango, accessed on August 10, 2025, <https://webtango.berkeley.edu/papers/thesis/chap2.pdf>
40. Best Computer Science Conferences Ranking 2024 | Research.com, accessed on August 10, 2025, <https://research.com/conference-rankings/computer-science>
41. Regional Conferences - Cal Poly Pomona, accessed on August 10, 2025, <https://www.cpp.edu/our-cpp/events-workshops/regional-conferences.shtml>
42. Top Computer Science Conferences & Events, accessed on August 10, 2025, <https://www.computer.org/conferences/top-computer-science-events/>
43. Exploring Sentiment Analysis in Indian Regional Languages: Methods, Challenges, and Future Directions | IJSREM Journal, accessed on August 10, 2025, <https://ijsrem.com/download/exploring-sentiment-analysis-in-indian-regional-languages-methods-challenges-and-future-directions/>
44. Emotion Detection and Sentiment Analysis in Regional Languages-A Review, accessed on August 10, 2025, <https://online.bamu.ac.in/naac_ssr/file_upload/28_32866_1414.pdf>
45. (PDF) Recent Advances in Sentiment Analysis of Indian Languages, accessed on August 10, 2025, <https://www.researchgate.net/publication/345240744_Recent_Advances_in_Sentiment_Analysis_of_Indian_Languages>
46. Top 60 Marathi News Websites in 2025 - Journalist Database - Feedspot, accessed on August 10, 2025, <https://journalists.feedspot.com/marathi_news_websites/>
47. Marathi News Paper Websites: A Webometric Study | 69198, accessed on August 10, 2025, <https://www.ijlis.org/abstract/marathi-news-paper-websites-a-webometric-study-69198.html>
48. Predicting Marathi News Class Using Semantic Entity-Driven Clustering Approach, accessed on August 10, 2025, <https://www.researchgate.net/publication/355310079_Predicting_Marathi_News_Class_Using_Semantic_Entity-Driven_Clustering_Approach>
49. IndiSentiment140: Sentiment Analysis Dataset for Indian Languages with Emphasis on Low-Resource Languages using Machine Translation - ACL Anthology, accessed on August 10, 2025, <https://aclanthology.org/2024.naacl-long.425/>
50. Aspect-Based Sentiment Analysis in Hindi Language by Ensembling ..., accessed on August 10, 2025, <https://www.mdpi.com/2079-9292/10/21/2641>