# **Memoir: The News Advance Project**

Author: Nacir Chahine

Date: August, 2025

Project: News Advance - An AI-Powered News Credibility Analyzer

## **Chapter 1: The Spark of an Idea - Dealing With the Information Problem**

Today's digital world has a strange problem. We have more information than ever, but finding the truth is getting harder. We see a constant flood of news, opinions, and stories, and it's tough to tell the difference between real journalism and fake news. This problem gets worse on social media, where algorithms show us what we already agree with, creating "filter bubbles" that keep us from seeing different points of view. The "News Advance" project started as a way to fix this.

Our mission was clear, but also very ambitious: to help the modern news reader. We know that being able to understand the media is a key skill today, so we wanted to build a tool that was more than just a news feed. We imagined a smart helper that would give people the confidence to look past the headlines and understand the complex world of online news. The main goal was to build a web app that uses Artificial Intelligence to help people check the articles they read, understand media bias, and spot potential fake news. By making things more transparent, we hoped to help create smarter readers who could have more meaningful conversations about important topics.

At its core, News Advance was designed to be a powerful analysis tool. The plan was for the system to collect articles from all kinds of news sites—from big, old newspapers to new online blogs—and run them through a set of tough AI checks. This included checking for political bias by looking for loaded words, and analyzing the emotional tone to see if an article was neutral or trying to make you feel a certain way. It would also create short, accurate summaries so people could get the main points quickly. Finally, we wanted to build a foundation for a fact-checking system to check claims against trusted sources. We didn't want to just build another news app; we wanted to build a new lens for people to see the news through, one that shows the structure, purpose, and quality of the content.

This memoir is the story of how we turned that idea into a real thing—a story about the tech we used, the code we wrote, the unexpected problems we faced, and the smart wins that made it all happen.

## **Chapter 2: The Blueprint - Designing a Modern Solution**

A project this complex needed a strong technical plan that was easy to manage and could grow over time. We chose every piece of technology for a specific reason, making sure our setup could handle what we needed now and what we might want to add later. Our philosophy was to use the best open-source tools available to build a system that was both powerful and flexible.

### **2.1 Core Technologies: The Backbone of the System**

* **Backend Framework:** We chose **Django 5.2**, a popular Python web framework famous for coming with a lot of built-in tools. Its Model-View-Template (MVT) design helped us keep our code organized, which was really important for a project with so many different parts. Django's powerful Object-Relational Mapper (ORM) hid the difficult SQL parts, letting us build and change our database structure quickly. On top of that, its built-in admin page and user login system saved us a huge amount of development time, and its security features protected us from common web attacks right from the start.
* **Programming Language:** **Python 3.8+** was the clear choice. Its simple code and huge number of available libraries made it the perfect language to connect our web app with our AI analysis tools. It let us smoothly combine everything into a single project.
* **Frontend Interface:** We built the user interface with standard **HTML5, CSS3, and JavaScript**, and we used the **Bootstrap 5** framework to make it look good. Bootstrap's powerful layout system and ready-to-use components let us build a responsive, mobile-first design that worked well on all devices, from phones to big screens, without needing to write a lot of custom CSS.
* **Data Gathering:** To keep our app filled with news, we used a mix of **Newspaper3k** and **BeautifulSoup4**. These tools were the heart of our data-gathering system. They helped us deal with the messy and unpredictable web, letting us grab, understand, and clean up articles from all kinds of websites, each with its own weird layout.

### **2.2 Database Structure: The Heart of the Data**

We used **SQLite** for development because it's simple and easy to get started with. For the live version of the app, we planned to switch to **PostgreSQL** to use its strength, ability to handle many users, and other cool features. We carefully organized our data into different models within three separate Django apps.

* **News Aggregator Models (news\_aggregator):** This app was the core of our content.
  + NewsSource: This stored key info about news publishers, including a reliability\_score that we planned to update over time based on how accurate they were.
  + NewsArticle: This was the most important model in the whole app. It held the article's content and information and was the center of everything, connecting to its source and its analysis results. Flags like is\_analyzed were key for managing our background processing system.
* **News Analysis Models (news\_analysis):** This app stored the results from our AI.
  + BiasAnalysis: This held the results of the political bias check. Using a scale (from left to right) instead of a simple "yes" or "no" gave us more detailed results, and the confidence score showed users how sure the AI was.
  + SentimentAnalysis: This stored the emotion scores. Breaking them down into positive, negative, and neutral parts let us show users more interesting visuals.
* **User & Profile Models (accounts):** This app took care of all the user-related features.
  + UserProfile: We extended the basic Django User model to add things like a user bio.
  + UserSavedArticle: This let users save articles they liked. We designed this not just as a bookmark feature, but also as a way to gather data for a future recommendation system.

With this strong data setup in place, we had built the vessel. The next, and most important, step was to make it work—to build the smart engine that would turn all this data into useful information.

## **Chapter 3: Building the Brains - AI, LLMs, and a Smart Pivot**

The real magic of News Advance was in its AI features. This is where we turned raw data into useful information. This chapter tells the story of how we built these features—a story of big successes where we planned, and tough challenges that needed us to change our plan and try new things.

### **3.1 A Custom-Trained Success: The Summarization Model**

One of our main goals was to build a custom summary model that could do better than standard tools. We chose the **BART** architecture, a powerful model that is perfect for generating text. We trained the facebook/bart-base model using the high-quality **BBC News Summary dataset**, which has thousands of news articles with summaries written by professional journalists.

The training process, which we managed with the Hugging Face Transformers library, was a careful process. The final model was a big win. It got good ROUGE scores (a way of measuring accuracy), but more importantly, it created summaries that made sense, fit the context, and were written in a news style. It successfully captured the main idea of an article without the choppy feeling of simpler methods. This success proved we could build and use our own custom AI solutions.

### **3.2 The Bias Detection Challenge and the LLM Pivot**

Our first thought was to do the same thing for political bias detection. But we quickly ran into a common problem in the AI world: there are no large, high-quality, and neutral datasets for political bias. We couldn't find any good data to train our model. The datasets that did exist were either too small, too specific, or were clearly biased themselves. Using them would just teach our model to copy that same bias, which would have been a deal-breaker for our app.

This problem put a key feature at risk, so we made a big strategic change. Instead of giving up, we decided to accept the new way of doing things in AI and integrate **Ollama**.

**Ollama** is a game-changing tool that lets you run powerful, open-source Large Language Models (LLMs) like Llama3 on your own machine. This new approach was a huge improvement and offered several big advantages:

1. **Nuance and Context:** LLMs have a smarter, more human-like understanding of language. They can spot subtle hints, persuasive language, and framing that simpler models would miss completely.
2. **Flexibility and Future-Proofing:** We could easily set up the system to use different LLMs through Ollama. This meant we weren't stuck with one model; we could keep improving our analysis just by switching to the newest and best open-source LLM.
3. **Explanatory Power:** A simple classifier just gives you a label like "left-leaning." An LLM, on the other hand, can be asked to explain *why* it made its decision. This "explainability" was a huge win for user trust.
4. **Privacy and Cost-Effectiveness:** By running the models locally, all user data and analysis stayed private on our servers. This also saved us from the high and unpredictable costs of using other companies' APIs.

This pivot wasn't a fix because we failed; it was a smart move ahead. It showed our team could adapt and recognized that the best solution isn't always the one you planned for. It let us deliver a bias analysis feature that was much better than we had first imagined.

## **Chapter 4: Execution and Results - A Hybrid AI in Action**

The execution phase was where our plans and strategies became a real, working app. The result wasn't just a list of features, but a single, hybrid AI system that smartly combined the power of our specialized, custom-trained model with the wide-ranging analytical abilities of general-purpose LLMs. This combination became the project's key technical feature.

* **High-Quality, Efficient Summarization:** Our custom BART model fit perfectly into our analysis pipeline. It gave users great summaries that were consistently more relevant and easier to read than those from simpler tools. Because it was a specialized model, it was also very fast, which was important for making the user experience feel snappy.
* **Nuanced Bias and Sentiment Analysis:** The Ollama integration was a huge success. By sending article text to a local Llama3 model with smart, carefully written prompts, the system could provide a very detailed analysis. For instance, instead of just saying an article was "negative," it could identify the main emotion as "outrage" or "fear." Instead of just a "center-left" label, it could explain *why*, noting that "the article uses words favored by one side and mostly quotes sources that support that viewpoint." This gave users a level of transparency and insight that a simple classifier never could.
* **A Unified, Insightful User Experience:** The real power of the system was in how we brought all these different analyses together into a clean and simple user interface. On one page, a user could see an article, its AI-generated summary, a visual of its political bias on a spectrum, and a color-coded breakdown of its emotional tone. This complete picture allowed a user to understand a piece of news in seconds, turning them from someone who just reads into someone who thinks critically about what they're reading. The project delivered on its main promise: to give users advanced, AI-powered tools to check the news they consume.

## **Chapter 5: Conclusion and Future Horizons**

Did the News Advance project reach its goal? Yes, absolutely. We started out wanting to build a smart tool to help people navigate the dangerous waters of the modern information world, and we did it. The platform is a powerful proof of concept that clearly shows how AI can be used not to replace human thinking, but to help it, leading to better media literacy and a stronger ability to spot misinformation.

The strategic pivot to LLMs with Ollama was a key moment for the project. It was a practical, forward-thinking decision that not only solved a big problem but also put our app on the cutting edge of AI. It turned a potential failure into one of our biggest strengths.

While the app is powerful and complete today, it's also just the beginning. The journey isn't over. Our future plans are ambitious, and we want to expand News Advance to make it an even more essential tool for the modern reader:

* **Advanced NLP Models & Granular Analysis:** We plan to train models to do even more detailed analysis, like identifying specific logical fallacies (e.g., personal attacks) or common propaganda tricks.
* **Comprehensive Fact-Checking:** A top priority is to build out our fact-checking feature. This means creating a system that can automatically pull out key claims from an article and check them against trusted fact-checking sites like PolitiFact, Snopes, and Reuters in real time.
* **API Development:** To increase our impact, we'll create a RESTful API. This will let other news apps, researchers, and developers use our analysis tools in their own projects, helping us spread our mission.
* **Content Expansion:** The next step is to go beyond text. We plan to add tools to analyze video and audio content, as well as the mix of images, memes, and text on social media.
* **Performance and Scalability:** As the app gets bigger, we'll need a stronger infrastructure. We'll use tools like Celery to run big analysis jobs in the background without slowing down the site, and Redis to cache results so they load instantly.

In the end, the News Advance project is more than just a web app. It's a real step toward a future where technology is a tool that makes things clearer, not more confusing. It empowers users to be smarter, more critical, and better-informed participants in the digital world. It's proof of what you can build with a clear idea, a solid technical plan, and the flexibility to adapt to new ideas in a fast-changing field—all to support a more informed democracy.