**Title:**  
**Smart-News App**

**Authors:**  
Priya Dumka

Bachelor of Technology  
Graphic Era Hill University, India  
Email: [Priyadumka03@gmail.com](mailto:Priyadumka03@gmail.com)

**Authors:**  
Pankaj Bisht

Bachelor of Technology  
Graphic Era Hill University, India

**Abstract**  
In today’s digital era, readers are overwhelmed by an abundance of online news sources, many of which may be unreliable or repetitive. To address these challenges, we present *SmartNews*, an AI-powered news aggregation platform that fetches, summarizes, and evaluates the credibility of news articles in real-time. Leveraging NLP techniques, transformer-based summarization models, and credibility scoring mechanisms using TF-IDF and cosine similarity, the system provides users with concise and reliable news tailored to their interests. The system supports modular integration with popular news APIs like GNews and NewsAPI and stores refined data in MongoDB.

**Keywords**: News Aggregation, AI, Summarization, Credibility Detection, Flask API, MongoDB, TF-IDF, Transformers

**1. Introduction**

The proliferation of digital media has led to both opportunities and challenges in news consumption. With misinformation and biased articles on the rise, there is an urgent need for intelligent systems that filter, condense, and verify news content. *SmartNews* addresses this need by aggregating news from multiple APIs, generating human-like summaries using deep learning, and assessing credibility via similarity comparison with trusted RSS feeds.

**2. System Architecture**

**2.1 Components**

* **Backend Server**: Flask-based REST API with endpoints for news retrieval, summarization, and favorite topic management.
* **News Sources**: GNews and NewsAPI for real-time news.
* **Database**: MongoDB for storing news articles by category.
* **Summarization Engine**: T5-base transformer model.
* **Credibility Engine**: Uses TF-IDF vectorization and cosine similarity with Google RSS feed articles.

**2.2 Workflow Overview**

1. User requests news via /api/news.
2. Articles are fetched from APIs and matched against RSS sources.
3. Credibility is computed based on content similarity and recency.
4. Summaries are generated using a fine-tuned NLP model.
5. Top credible articles are stored and served.

**3. Technical Stack**

* **Language**: Python
* **Libraries**: Flask, transformers, scikit-learn, nltk, requests, pymongo, xml.etree, email.utils
* **Model**: t5-base from Hugging Face Transformers
* **Database**: MongoDB
* **Web Framework**: Flask with CORS enabled
* **News APIs**: NewsAPI, GNews
* **Deployment**: Flask app running on 0.0.0.0, suitable for Docker or server environments

**4. Summarization Methodology**

We use the T5-base transformer to convert lengthy article content into concise summaries. The input to the model is prefixed with "summarize:" and truncated to 512 words if necessary. The model outputs a summary between 80–130 words using deterministic decoding (do\_sample=False).

**5. Credibility Scoring Algorithm**

Credibility is derived from:

* **Content Similarity**: Using TF-IDF vectors and cosine similarity between the article and RSS feeds.
* **Recency**: Bonus scoring for articles published in the last 2–7 days.
* **Source Diversity**: Additional weight for multiple trusted domains.
* **Clickbait Detection**: Penalty for titles matching clickbait patterns.

Final score is computed as:

ini

CopyEdit

score = base\_score + diversity\_score + recency\_score - clickbait\_penalty

**6. Use Cases**

* **Smart Feed**: Fetches credible and relevant news in real-time.
* **Favorite Topics**: Stores and serves user-preferred categories like Technology, Business, Science, etc.
* **Fake News Filtering**: Reduces exposure to untrustworthy sources by comparing with RSS data.

**7. Conclusion and Future Work**

SmartNews successfully integrates AI-driven summarization and credibility analysis to enhance the user news experience. Future enhancements include multilingual support, real-time user feedback on article quality, UI/UX improvements, and integration with mobile apps.

**References**

1. Raffel, C., et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *JMLR*, 2020.
2. NewsAPI: <https://newsapi.org>
3. GNews: <https://gnews.io>
4. Hugging Face Transformers: https://huggingface.co/transformers

**Abstract**  
In today’s digital era, readers are overwhelmed by an abundance of online news sources, many of which may be unreliable or repetitive. To address these challenges, we present *SmartNews*, an AI-powered news aggregation platform that fetches, summarizes, and evaluates the credibility of news articles in real-time. Leveraging NLP techniques, transformer-based summarization models, and credibility scoring mechanisms using TF-IDF and cosine similarity, the system provides users with concise and reliable news tailored to their interests. The system supports modular integration with popular news APIs like GNews and NewsAPI and stores refined data in MongoDB.

**Keywords**: News Aggregation, AI, Summarization, Credibility Detection, Flask API, MongoDB, TF-IDF, Transformers

**1. Introduction**

The proliferation of digital media has led to both opportunities and challenges in news consumption. With misinformation and biased articles on the rise, there is an urgent need for intelligent systems that filter, condense, and verify news content. *SmartNews* addresses this need by aggregating news from multiple APIs, generating human-like summaries using deep learning, and assessing credibility via similarity comparison with trusted RSS feeds.

**2. System Architecture**

**2.1 Components**

* **Backend Server**: Flask-based REST API with endpoints for news retrieval, summarization, and favorite topic management.
* **News Sources**: GNews and NewsAPI for real-time news.
* **Database**: MongoDB for storing news articles by category.
* **Summarization Engine**: T5-base transformer model.
* **Credibility Engine**: Uses TF-IDF vectorization and cosine similarity with Google RSS feed articles.

**2.2 Workflow Overview**

1. User requests news via /api/news.
2. Articles are fetched from APIs and matched against RSS sources.
3. Credibility is computed based on content similarity and recency.
4. Summaries are generated using a fine-tuned NLP model.
5. Top credible articles are stored and served.

**3. Technical Stack**

* **Language**: Python
* **Libraries**: Flask, transformers, scikit-learn, nltk, requests, pymongo, xml.etree, email.utils
* **Model**: t5-base from Hugging Face Transformers
* **Database**: MongoDB
* **Web Framework**: Flask with CORS enabled
* **News APIs**: NewsAPI, GNews
* **Deployment**: Flask app running on 0.0.0.0, suitable for Docker or server environments

**4. Summarization Methodology**

We use the T5-base transformer to convert lengthy article content into concise summaries. The input to the model is prefixed with "summarize:" and truncated to 512 words if necessary. The model outputs a summary between 80–130 words using deterministic decoding (do\_sample=False).

**5. Credibility Scoring Algorithm**

Credibility is derived from:

* **Content Similarity**: Using TF-IDF vectors and cosine similarity between the article and RSS feeds.
* **Recency**: Bonus scoring for articles published in the last 2–7 days.
* **Source Diversity**: Additional weight for multiple trusted domains.
* **Clickbait Detection**: Penalty for titles matching clickbait patterns.

Final score is computed as:

ini

CopyEdit

score = base\_score + diversity\_score + recency\_score - clickbait\_penalty

**6. Use Cases**

* **Smart Feed**: Fetches credible and relevant news in real-time.
* **Favorite Topics**: Stores and serves user-preferred categories like Technology, Business, Science, etc.
* **Fake News Filtering**: Reduces exposure to untrustworthy sources by comparing with RSS data.

**7. Conclusion and Future Work**

SmartNews successfully integrates AI-driven summarization and credibility analysis to enhance the user news experience. Future enhancements include multilingual support, real-time user feedback on article quality, UI/UX improvements, and integration with mobile apps.

**References**

1. Raffel, C., et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *JMLR*, 2020.
2. NewsAPI: <https://newsapi.org>
3. GNews: <https://gnews.io>
4. Hugging Face Transformers: https://huggingface.co/transformers