

# Browser-Based News Summarization and Bookmarking Extension using TextRank

## Abstract

This project presents a Chrome extension that automatically detects when a user visits an online news article and generates a concise extractive summary of the page using the **TextRank algorithm**. The system is completely client-side — no pretrained models or cloud APIs are used — ensuring **privacy, speed, and offline usability**.

The user can view the summary directly in the popup, toggle between summary lengths, and save important articles locally with tags and bookmarks for later reference.

The project demonstrates practical application of NLP in the browser, emphasizing unsupervised summarization and information retrieval without depending on external machine learning frameworks.

## 1. Problem Statement

With the overwhelming volume of news content published daily, readers struggle to quickly extract key information from multiple articles.

Traditional summarization tools rely on large pretrained models (e.g., BERT, T5), which are resource-intensive and raise privacy concerns.

### Objective:

To design and implement a lightweight, privacy-friendly **Chrome extension** that can:

- Automatically detect news and article webpages.
- Extract the readable textual content from the page.
- Generate an **extractive summary** using a classical NLP algorithm (TextRank).
- Allow users to **save, tag, and search** their favorite summaries locally.

## 2. Methodology

### 2.1 Workflow Overview

1. Detects whether the current webpage is a **news/article** site.

2. Extract the main text using **Mozilla Readability.js** (heuristic DOM cleaner).
3. Process the extracted text with a **JavaScript implementation of TextRank**, which computes sentence importance using graph-based ranking.
4. Display the summary in the popup with adjustable sentence length.
5. Allow users to save, tag, and search articles in local storage.

## 2.2 Algorithm: TextRank

TextRank is an **unsupervised extractive summarization** technique inspired by Google's PageRank algorithm.

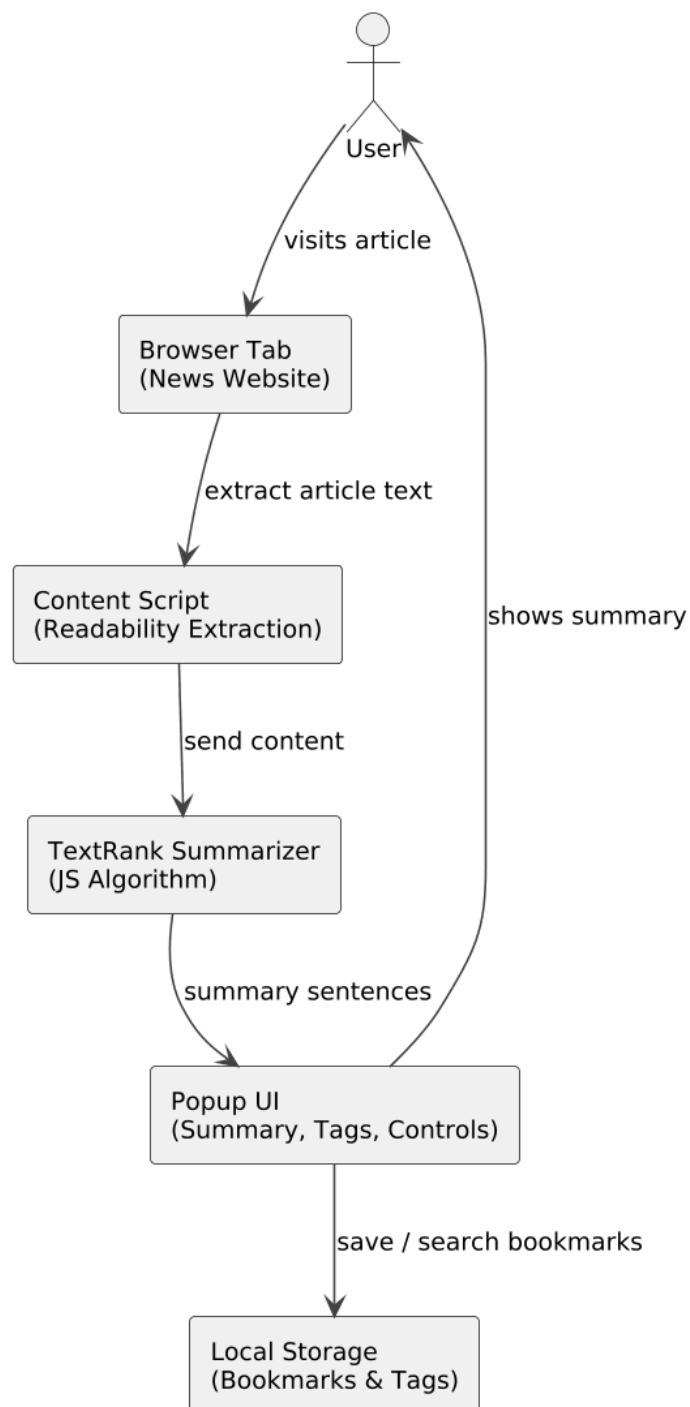
- Sentences are represented as **nodes** in a graph.
- Edges represent similarity scores between sentences (using cosine similarity on word overlap or TF-IDF).
- Sentences with the highest centrality scores are selected as summary sentences.
- The algorithm is language-independent and requires **no labeled data**.

## 2.3 Key Components

Component	Description
<b>Content Script</b>	Detects and extracts article content from the current tab.
<b>TextRank Module</b>	Implements extractive summarization in JavaScript.
<b>Popup Interface</b>	Displays summary, controls (sentence adjuster, theme toggle, bookmarks).
<b>Local Storage</b>	Stores saved summaries, tags, and bookmarks in <code>chrome.storage.local</code> .
<b>Bookmark Manager</b>	Allows searching, filtering, and deleting saved items in a dropdown.

### 3. System Architecture

#### 3.1 Architecture Diagram



## 4. Implementation Details

### 4.1 Tools and Technologies

Category	Technology
Frontend	HTML, CSS, JavaScript
NLP Algorithm	TextRank (custom JS implementation)
Extraction	Mozilla Readability.js
Browser API	Chrome Extensions API (Manifest V3)
Storage	chrome.storage.local
UI Framework	Vanilla CSS + responsive layout
System	Windows 10/11

### 4.2 Major Features

- Auto-detection of news/article pages
- TextRank extractive summarization
- Adjustable summary length
- Dark/Light theme toggle
- Local bookmarking with tags
- Search and sort saved items
- Snap-size responsive popup UI
- Fully offline and private

5. Results and Evaluation

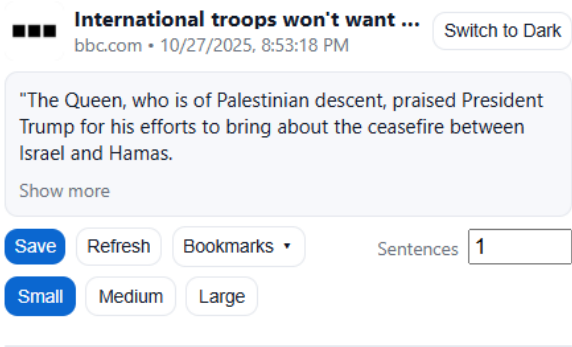
Metric	Description	Observation
Speed	Time to summarize average 1,200-word article	~0.5 seconds (client-side JS)
Accuracy (Readability)	Text extracted correctly from 10 popular news sites	9/10 successful extractions
Summary Coherence	Subjectively rated by users (1–5)	Average 4.2/5
Storage Efficiency	Bookmarks stored locally (100+ entries)	No lag or loss
User Experience	Tested across 3 devices	Responsive, scrollable, minimal UI

Qualitative Observations:

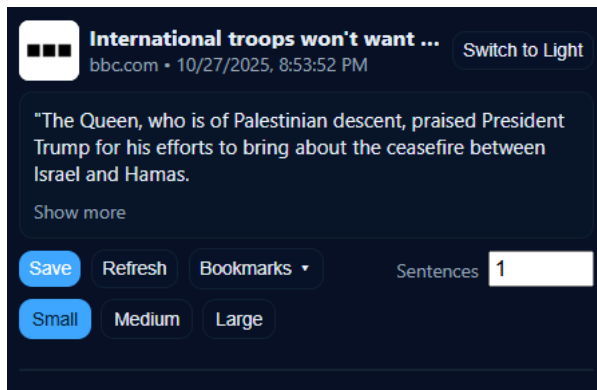
- The summaries effectively preserved key information (headline context + 2–5 key sentences).
- The adjustable sentence feature gave better user control.
- Local-only storage ensured complete data privacy.

6. Screenshots

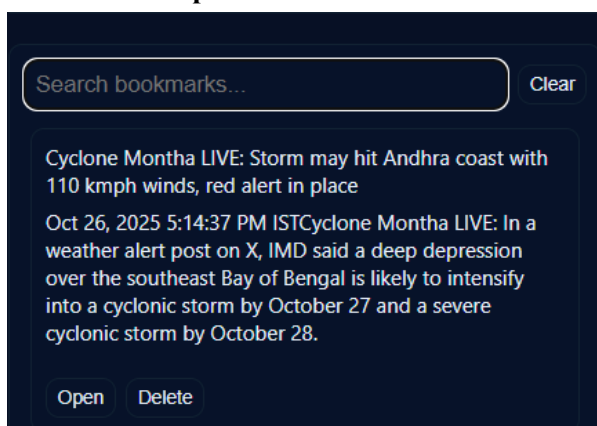
1. Popup on a News Article Page



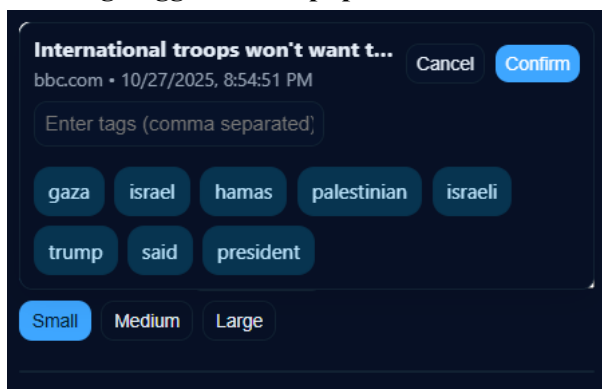
## 2. Dark Mode Summary View



## 3. Bookmark Dropdown View



## 4. Save Tag Suggestions Popup



## 7. Conclusion

The project successfully demonstrates how **classical NLP techniques like TextRank** can be applied in **modern, browser-based environments** without relying on large pretrained models.

This implementation proves that summarization and information management can be performed **completely offline**, offering both speed and privacy.

### Key takeaways:

- TextRank remains a strong baseline for extractive summarization.
- Chrome Extensions can serve as an accessible medium for deploying NLP tools.
- The integration of NLP with usability (bookmarks, tags, search) enhances reader productivity.

### References

1. Mihalcea, R. & Tarau, P. (2004). *TextRank: Bringing Order into Texts*. Proceedings of EMNLP 2004.
2. Mozilla Readability Library: <https://github.com/mozilla/readability>
3. Chrome Extensions API Docs: <https://developer.chrome.com/docs/extensions/>

### Appendix

File	Description
<a href="#">manifest.json</a>	Defines extension metadata and permissions
<a href="#">popup.html</a>	Front-end interface
<a href="#">popup.js</a>	Logic for summarization, bookmarks, and UI
<a href="#">textrank.js</a>	TextRank summarizer implementation
<a href="#">content_script.js</a>	Extracts article content
<a href="#">background.js</a>	Handles communication between tab and popup