const markdown = $input.first().json.markdown\_content || '';

const lines = markdown.split('\n');

// Footer / boilerplate indicators that mark lines to exclude entirely

const excludeLineKeywords = [

'Follow us on',

'Found this article interesting?',

'Subscribe – Get Latest News',

'Home',

'Newsletter',

'Webinars',

'Expert Insights',

'Contact',

'Resources',

'About THN',

'Jobs',

'Advertise with us',

'Contact/Tip Us',

'RSS Feeds',

'Email Alerts',

'javascript:void(0)',

'blob:http://localhost',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'[\_\_]',

'\_\_',

'\_\_',

'\_\_',

'\_\_',

'\_\_',

'\_\_',

'\_\_',

'===============',

'==========================================================================================================================================================================',

'---------------------------------------------------------------',

];

// Domains to block for URLs inside lines (remove URLs containing these domains)

const blockedDomains = [

'blogger.googleusercontent.com',

'blogger.com',

];

// Remove markdown images

function removeImages(line) {

return line.replace(/\[!\[.\*?\]\(.\*?\)\]\(.\*?\)/g, '').replace(/!\[.\*?\]\(.\*?\)/g, '');

}

// Convert markdown links [text](url) to just text

function removeLinks(line) {

return line.replace(/\[([^\]]+)\]\((https?:\/\/[^\s)]+)\)/g, '$1');

}

// Remove URLs from blocked domains inside the line

function removeBlockedUrls(line) {

return line.replace(/https?:\/\/[^\s)]+/g, url => {

for (const domain of blockedDomains) {

if (url.includes(domain)) return ''; // remove URL completely

}

return url; // keep other URLs

}).replace(/\s{2,}/g, ' ').trim();

}

const filteredLines = [];

for (let line of lines) {

line = line.trim();

if (!line) {

// Treat empty line as paragraph separator, keep it

filteredLines.push('');

continue;

}

if (excludeLineKeywords.some(keyword => line.includes(keyword))) {

continue; // skip excluded lines

}

line = removeImages(line);

line = removeLinks(line);

line = removeBlockedUrls(line);

if (line.length > 0) {

filteredLines.push(line);

}

}

// Now join lines into a single string, replacing all single \n with space to remove line breaks \*inside\* paragraphs

// and keeping double \n\n as paragraph breaks.

// To do this, first join by '\n' so paragraphs are separated by empty lines,

// then replace all remaining single newlines inside paragraphs with spaces.

// We'll do this by joining with '\n', then replacing any '\n' that is NOT part of double '\n\n' with space.

let cleanedContent = filteredLines.join('\n');

// Replace all single \n (not double) with space

cleanedContent = cleanedContent.replace(/([^\n])\n([^\n])/g, '$1 $2');

// Finally trim

cleanedContent = cleanedContent.trim();

return [{

json: {

title: $input.first().json.title,

url: $input.first().json.url,

published\_time: $input.first().json.published\_time,

cleaned\_text: cleanedContent,

}

}];