Extracting Structured Insights from Hotel Websites: Comparing Traditional Python Methods with LLM-Powered Approaches

Team: Quantitative Visionaries

Members: Yuyan Bei, Yicheng Diao, Zheyu Zheng, Zhenyi Zhao

I. Research Background and Practical Relevance

This study investigates how traditional Python-based methods compare to large language model (LLM)-powered approaches in extracting structured insights from unstructured textual data in the online travel industry. The ability to analyze hotel availability and pricing across OTAs provides investors with valuable intelligence on platform competitiveness and regional strategies—particularly in the Asia-Pacific market, where players like Trip.com (TCOM), Booking.com (BKNG), and Expedia (EXPE) are highly active. These data variations serve as alternative indicators for evaluating market share, pricing power, and demand trends, ultimately supporting more informed equity research and investment decisions

Our motivation stems from a real-world case during one team member's internship, where they were tasked with assessing the relative strengths of OTA brands in Asia-Pacific. Specifically, the goal was to determine which platforms offered broader hotel coverage and more competitive pricing. Since field research across multiple countries is infeasible, analysts must rely heavily on OTA search results as a key source of intelligence. Previously, such analysis was based on small-sample manual reviews, a method prone to bias and lacking scalability.

To address these limitations, the team member developed a Python-based pipeline that used regular expressions and rule-based parsing to extract structured information from OTA web pages. This represented a meaningful improvement over manual sampling but still faced challenges in efficiency, adaptability, and information completeness. These limitations point to the potential of AI to enhance both the speed and accuracy of such workflows.

II. Methodological Gap and Motivation for AI Integration

Unstructured hotel listings on OTA platforms contain critical data for comparative analysis, market research, and pricing intelligence. However, their diverse formats, inconsistent structures, and multilingual content present considerable challenges for rule-based data processing. Traditional tools—such as BeautifulSoup, re, or pandas—work reasonably well in narrowly defined, well-structured environments but often fail when applied to dynamic, messy, or customized datasets.

This is where LLM-based solutions offer a significant advantage. By leveraging the natural language understanding and flexible parsing capabilities of AI, we can more effectively navigate the heterogeneity of OTA data. These models can identify and extract relevant information without the need for extensive pre-programmed rules, making them ideal for tasks where input structure varies across sources.

The objective of this project is to systematically compare the performance of traditional Python-based techniques with AI-powered methods in processing real-world OTA data. Specifically, we assess their ability to extract key variables—such as hotel brand presence by region and price competitiveness among overlapping listings—and evaluate the trade-offs in terms of accuracy, completeness, and scalability. In doing so, we aim to determine whether LLM-powered tools can meaningfully improve the quality and practicality of OTA-based investment research.

III. **Data and Sources**

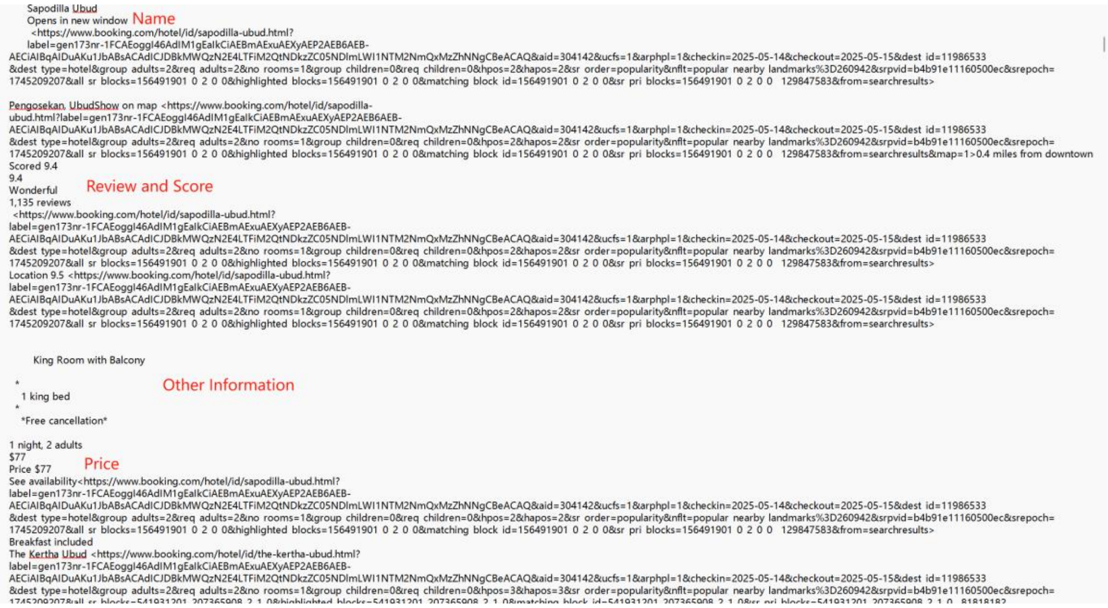
To conduct our analysis, we collected hotel listing data from three major online travel agency (OTA) platforms: Trip.com (TCOM), a leading China-based travel booking site; Booking.com (BKNG), a global leader in accommodation reservations; and Expedia (EXPE), a U.S.-based travel agency with a strong international presence. To ensure geographic diversity while keeping the dataset focused and manageable, we selected four representative tourist locations in the Asia-Pacific region: downtown Siem Reap in Cambodia, Sacred Monkey Forest Park in Bali, Rizal Park in Manila, and the Osaka Castle area in Japan.

One of the main challenges in data collection was the inherent resistance of these platforms to automated scraping. Most OTAs use dynamic front-end rendering powered by JavaScript to populate key content, making such content inaccessible to standard scraping tools. Furthermore, these platforms often deploy anti-scraping mechanisms, including frequent changes to DOM structure, rate-limiting, and IP blocking. These measures make large-scale extraction using conventional tools like requests and BeautifulSoup both unreliable and impractical.

To overcome these limitations, we adopted a controlled manual extraction strategy using a safe and consistent automated process. This method involves programmatically loading all relevant webpage content and saving it locally as a .txt file. By doing so, we were able to collect a high-quality, uniform dataset suitable for both traditional text processing and LLM-assisted analysis. This initial extraction step was kept identical across both methodological pipelines to ensure a fair and consistent basis for comparison.

As illustrated in the image below using Booking.com as an example, hotel listing data extracted into .txt files appears highly unstructured and messy. The scattered format, inconsistent labeling, and lack of clear delimiters make manual information extraction nearly impossible. In such contexts, programming becomes essential. However, the diversity in how different platforms structure their search result pages introduces additional complexity: traditional Python-based text extraction requires custom, case-by-case programming tailored to the unique formatting rules of each OTA.

Figure 1. Sample Raw Text Output of Hotel Listings from Booking.com



```
    Sapodilla Ubud
    Opens in new window
     <https://www.booking.com/hotel/id/sapodilla-ubud.html?
    label=gen173nr-1FCAEoggI46AdIM1gEaIkCiAEBmAExuAEXyAEP2AEB6AEB-
AECiAIBqAIDuAKu1JbABsACAdICJDBkMWQzN2E4LTFiM2QtNDkzZC05NDlmLWI1NTM2NmQxMzZhNNgCBeACAQ&aid=304142&ucfs=1&arphpl=1&checkin=2025-05-14&checkout=2025-05-15&dest id=11986533
&dest type=hotel&group adults=2&req adults=2&no rooms=1&group children=0&req children=0&hpos=2&hapos=2&sr order=popularity&nflt=popular nearby landmarks%3D260942&srpvid=b4b91e11160500ec&srepoch=
1745209207&all sr blocks=156491901 0 2 0 0&highlighted blocks=156491901 0 2 0 0&matching block id=156491901 0 2 0 0&sr pri blocks=156491901 0 2 0 0  129847583&from=searchresults>

Pengosekan, UbudShow on map <https://www.booking.com/hotel/id/sapodilla-
ubud.html?label=gen173nr-1FCAEoggI46AdIM1gEaIkCiAEBmAExuAEXyAEP2AEB6AEB-
AECiAIBqAIDuAKu1JbABsACAdICJDBkMWQzN2E4LTFiM2QtNDkzZC05NDlmLWI1NTM2NmQxMzZhNNgCBeACAQ&aid=304142&ucfs=1&arphpl=1&checkin=2025-05-14&checkout=2025-05-15&dest id=11986533
&dest type=hotel&group adults=2&req adults=2&no rooms=1&group children=0&req children=0&hpos=2&hapos=2&sr order=popularity&nflt=popular nearby landmarks%3D260942&srpvid=b4b91e11160500ec&srepoch=
1745209207&all sr blocks=156491901 0 2 0 0&highlighted blocks=156491901 0 2 0 0&matching block id=156491901 0 2 0 0&sr pri blocks=156491901 0 2 0 0  129847583&from=searchresults&map=1>0.4 miles from downtown
Scored 9.4
9.4
Wonderful
1,135 reviews
 <https://www.booking.com/hotel/id/sapodilla-ubud.html?
label=gen173nr-1FCAEoggI46AdIM1gEaIkCiAEBmAExuAEXyAEP2AEB6AEB-
AECiAIBqAIDuAKu1JbABsACAdICJDBkMWQzN2E4LTFiM2QtNDkzZC05NDlmLWI1NTM2NmQxMzZhNNgCBeACAQ&aid=304142&ucfs=1&arphpl=1&checkin=2025-05-14&checkout=2025-05-15&dest id=11986533
&dest type=hotel&group adults=2&req adults=2&no rooms=1&group children=0&req children=0&hpos=2&hapos=2&sr order=popularity&nflt=popular nearby landmarks%3D260942&srpvid=b4b91e11160500ec&srepoch=
1745209207&all sr blocks=156491901 0 2 0 0&highlighted blocks=156491901 0 2 0 0&matching block id=156491901 0 2 0 0&sr pri blocks=156491901 0 2 0 0  129847583&from=searchresults>
Location 9.5 <https://www.booking.com/hotel/id/sapodilla-ubud.html?
label=gen173nr-1FCAEoggI46AdIM1gEaIkCiAEBmAExuAEXyAEP2AEB6AEB-
AECiAIBqAIDuAKu1JbABsACAdICJDBkMWQzN2E4LTFiM2QtNDkzZC05NDlmLWI1NTM2NmQxMzZhNNgCBeACAQ&aid=304142&ucfs=1&arphpl=1&checkin=2025-05-14&checkout=2025-05-15&dest id=11986533
&dest type=hotel&group adults=2&req adults=2&no rooms=1&group children=0&req children=0&hpos=2&hapos=2&sr order=popularity&nflt=popular nearby landmarks%3D260942&srpvid=b4b91e11160500ec&srepoch=
1745209207&all sr blocks=156491901 0 2 0 0&highlighted blocks=156491901 0 2 0 0&matching block id=156491901 0 2 0 0&sr pri blocks=156491901 0 2 0 0  129847583&from=searchresults>


     King Room with Balcony

  *
   1 king bed
  *
   *Free cancellation*

1 night, 2 adults
$77
Price $77
See availability<https://www.booking.com/hotel/id/sapodilla-ubud.html?
label=gen173nr-1FCAEoggI46AdIM1gEaIkCiAEBmAExuAEXyAEP2AEB6AEB-
AECiAIBqAIDuAKu1JbABsACAdICJDBkMWQzN2E4LTFiM2QtNDkzZC05NDlmLWI1NTM2NmQxMzZhNNgCBeACAQ&aid=304142&ucfs=1&arphpl=1&checkin=2025-05-14&checkout=2025-05-15&dest id=11986533
&dest type=hotel&group adults=2&req adults=2&no rooms=1&group children=0&req children=0&hpos=2&hapos=2&sr order=popularity&nflt=popular nearby landmarks%3D260942&srpvid=b4b91e11160500ec&srepoch=
1745209207&all sr blocks=156491901 0 2 0 0&highlighted blocks=156491901 0 2 0 0&matching block id=156491901 0 2 0 0&sr pri blocks=156491901 0 2 0 0  129847583&from=searchresults>
Breakfast included
The Kertha Ubud <https://www.booking.com/hotel/id/the-kertha-ubud.html?
label=gen173nr-1FCAEoggI46AdIM1gEaIkCiAEBmAExuAEXyAEP2AEB6AEB-
AECiAIBqAIDuAKu1JbABsACAdICJDBkMWQzN2E4LTFiM2QtNDkzZC05NDlmLWI1NTM2NmQxMzZhNNgCBeACAQ&aid=304142&ucfs=1&arphpl=1&checkin=2025-05-14&checkout=2025-05-15&dest id=11986533
&dest type=hotel&group adults=2&req adults=2&no rooms=1&group children=0&req children=0&hpos=3&hapos=3&sr order=popularity&nflt=popular nearby landmarks%3D260942&srpvid=b4b91e11160500ec&srepoch=
1745209207&all sr blocks=541931201 207365908 2 1 0&highlighted blocks=541931201 207365908 2 1 0&matching block id=541931201 207365908 2 1 0&sr pri blocks=541931201 207365908 2 1 0  81818182
```

By contrast, when leveraging AI models in combination with Python scripts, the extraction process becomes significantly more efficient and flexible. Rather than designing separate parsing logic for each case, we can simply send a batch of text blocks—each containing raw hotel listing content—to an LLM-powered interface. The AI is capable of interpreting this unstructured text and intelligently extracting key variables, such as hotel names, brands, prices, and distances. This adaptability reduces development time and improves consistency across platforms, making LLM-based methods a promising alternative to traditional rule-based parsing in real-world, heterogeneous data environments.

IV. Methodology: Traditional Parsing vs. LLM-Based Information Extraction

In this study, we employed and compared two distinct methodologies for processing unstructured textual data extracted from OTA websites: a traditional Python-based parsing approach and a more advanced LLM-assisted method.

4.1 Traditional Python-Based Parsing

The traditional method relied on Python tools such as regular expressions, keyword matching, and manual string operations. Due to the varying HTML and text structures across different OTAs, this approach required building customized parsing logic for each platform. While relatively efficient for small-scale, consistent datasets, its limitations became evident when handling large-scale, messy, or multilingual content.

Regex-based parsing lacked semantic understanding and proved brittle—small changes in formatting, punctuation, or mixed-language content could cause parsing failures. Moreover, maintaining individual pipelines for each OTA became time-consuming and error-prone. One critical challenge arose in identifying and separating hotel information blocks: when a hotel name began with non-English characters (e.g., symbols, numerals, or non-Latin scripts), the parser often failed to recognize it, causing its details to be misclassified or merged with neighboring entries. Although infrequent, such cases compromised the accuracy of the final structured data.

4.2 LLM-Powered Information Extraction via DeepSeek

In contrast, our primary approach leveraged a large language model (LLM) to extract structured data with greater robustness and generalizability. Specifically, we used the DeepSeek API to access a GPT-like model capable of semantically parsing raw text blocks. Our choice of model was driven by cost-efficiency: DeepSeek offers extremely low usage costs, making it a practical solution for large-scale data extraction tasks. This consideration is especially important because traditional Python-based methods, while labor-intensive, incur no monetary cost. To ensure a fair comparison, we treated model affordability as a variable control—favoring a low-cost LLM solution that aligns more closely with the "free" nature of traditional parsing.

In terms of technical implementation, we first applied minimal regex to segment the raw .txt content into individual hotel listing blocks. These text blocks were then passed into the LLM using a standardized system prompt. The prompt instructed the model to extract key hotel attributes—such as name, location, distance, rating, price, and amenities—and return them in a structured JSON format. Since the information to be extracted was relatively explicit and straightforward, this task fell well within the capabilities of even general-purpose LLMs.

Crucially, we maintained a consistent prompt across all OTA platforms, modifying only the name of the platform in the prompt, while keeping the overall structure and extraction instructions unchanged. This abstraction enabled a modular and highly adaptable workflow. In addition, careful tuning of the prompt—including temperature settings and clearly defined output instructions—was essential to prevent AI hallucinations and maintain accuracy. The low complexity of the extraction task, combined with effective prompt design, allowed us to leverage the LLM's semantic capabilities without overcomplicating the process or incurring significant cost.

This AI-powered pipeline offered clear advantages over traditional methods, particularly in its ability to handle inconsistent formatting, multilingual content, and subtle variations in how information is presented. Overall, the LLM-based approach delivered a more scalable and reliable solution for structuring unorganized hotel listing data across multiple OTA platforms.

For transparency and reproducibility, the implementation code for both methods will be presented in the appendix. Regardless of the extraction method used, the ultimate goal was the same: to transform unstructured web data into a consistent, analyzable format for downstream processing.

IV. Empirical Results

In evaluating the performance of the two approaches, we found that while traditional Python-based text analysis methods can, with extensive customization and iterative debugging, produce results broadly comparable to those derived from LLM-assisted extraction, they do so at a significantly higher time and labor cost. Specifically, even with AI assistance in code writing (limited to programming, not extraction), building traditional parsing pipelines for a single platform required more than three hours of manual effort. Moreover, any structural change to the OTA's front-end would require reprogramming substantial portions of the extraction logic. This fragility and lack of adaptability present serious limitations for maintaining long-term or multi-platform analyses.

In contrast, the AI-powered extraction approach demonstrated strong advantages in both efficiency and generalizability. The prompt design was consistent across platforms (with only the OTA name changed), and once the block segmentation logic was defined, the process required minimal platform-specific customization. This modularity and semantic flexibility allowed us to process a wide range of OTA content formats with little additional effort.

Using the AI-based pipeline, we generated structured datasets that closely mirrored those produced by traditional parsing. These structured outputs supported two key tasks in our comparative analysis:

Hotel Availability by Brand and Region: We identified the number of hotels affiliated with each brand across the four selected Asia-Pacific regions, allowing us to compare brand-level geographic coverage.

Relative Price Competitiveness: To assess pricing strategy, we performed fuzzy matching to determine whether the same hotel appeared under multiple brands. For each matched hotel, we computed a price competitiveness coefficient defined as the individual brand’s listed price divided by the average price for that hotel across all brands.

The AI-based analysis proved more capable of identifying complete and accurate hotel information, reducing the risk of omissions or misclassifications common in rule-based approaches. As a result, we were able to derive more reliable and nuanced estimates of pricing power and platform competitiveness.

Overall, the empirical evidence supports the conclusion that while traditional methods can approximate the desired outputs under tightly controlled conditions, LLM-assisted extraction offers a far more scalable and resilient alternative—particularly in real-world scenarios involving unstructured, inconsistent, and multilingual content.

Table 1. OTA Hotel Counts and Price Ratios: Traditional vs. AI Extraction

Country	Sampling Scope	Trip.com Hotels	Booking.com Hotels	Expedia Hotels	Trip.com Price	Booking.com Price	Expedia Price	Version
Japan (Osaka)	Hotels within 3 miles of Osaka Castle	170	524	350	1.01	0.99	1.09	Traditional
Cambodia (Siem Reap)	All hotels in city center	320	520	415	1.02	0.95	1.1	Traditional
Indonesia (Ubud)	All hotels near Sacred Monkey Forest Sanctuary	220	297	298	1.03	0.93	1.07	Traditional
Philippines (Manila)	All hotels near Rizal Park	220	297	298	1.03	0.93	1.07	Traditional
Japan (Osaka)	Hotels within 3 miles of Osaka Castle	194	603	367	0.93	1.06	1.02	AI
Cambodia (Siem Reap)	All hotels in city center	359	596	491	1.11	0.89	0.99	AI
Indonesia (Ubud)	All hotels near Sacred Monkey Forest Sanctuary	247	350	336	1.13	0.85	1.17	AI
Philippines (Manila)	All hotels near Rizal Park	237	326	334	1.1	0.88	1.14	AI

V. Comparative Evaluation of Traditional vs. LLM-Based Approaches

The limitations of the traditional parsing approach became particularly evident in edge cases and noisy inputs. While regular expressions and rule-based logic perform adequately under highly structured and static formats, they fail to generalize across variations. As a result, traditional methods often suffer from false positives and critical omissions, especially when listing structures differ across OTAs or are updated over time.

Multilingual content, inconsistent punctuation, and irregular formatting further exacerbate the brittleness of this rule-based approach. Moreover, the development and maintenance of scraping logic for each platform are time-intensive, and even small structural changes on a website can break an entire pipeline, necessitating costly revisions.

In contrast, the LLM-based extraction method—built on a unified prompt and applied across all OTA platforms—demonstrated a significantly more robust and intelligent approach to parsing. The model effectively transformed unstructured blocks of text into semantically accurate and consistently formatted JSON outputs. Crucially, the prompt design remained stable across platforms, requiring only minor tweaks (e.g., changing the platform name), while the model successfully adapted to differences in text flow, language, and structure.

The greatest strengths of the LLM approach lie in its **scalability**, **contextual awareness**, and **resilience to noise**. It can accommodate diverse formats, support rapid onboarding of new data sources, and adapt to real-world inconsistencies without the need for platform-specific logic. Even when listings contain irrelevant or cluttered content, the model focuses on meaning rather than surface patterns, maintaining performance where traditional parsing fails.

While the use of an LLM API does introduce latency and cost, these drawbacks are offset by significantly lower development time and maintenance overhead. The table below summarizes the core trade-offs:

Table 2. Comparative Summary of Traditional vs. LLM-Based Text Extraction Approaches

Criteria	Traditional Python Parsing	LLM-Based Extraction
Semantic Understanding	Absent – relies on patterns	Strong – interprets context and meaning
Adaptability to Format Changes	Low – highly fragile	High – prompt remains effective across formats
Multilingual Support	Poor – requires custom rules	Native support across languages
Development Time	High – platform-specific parsing logic	Low – reusable prompts with minor tuning
Maintenance Burden	High – breaks easily with updates	Low – resilient to noise and variation
Scalability	Limited	Excellent
Cost	Free (excluding developer time)	Minimal (API usage cost)
Latency per Task	Instantaneous	Delay due to API call
False Positives/Negatives	Common	Rare

VI. Insights and Conclusion

This project demonstrates that for tasks involving the transformation of unstructured or semi-structured web data into structured formats, LLM-based methods significantly outperform traditional Python parsing techniques. By enabling scalable, flexible, and semantically aware data extraction, LLMs reduce the complexity and manual effort involved in building and maintaining rule-based systems—particularly when working with noisy, multilingual, or inconsistent content such as OTA listings.

While our analysis pipeline ultimately produces structured datasets ready for empirical modeling (e.g., market competitiveness, price sensitivity, brand coverage), the focus of this project is not on downstream empirical analysis. There are two reasons for this deliberate choice. First, from a methodological standpoint, the core contribution of this work lies in using AI to transform alternative data—textual web content—into structured, numerical data. Once this transformation is complete, subsequent analysis falls into the domain of conventional structured-data analytics, which is not the focus of this course.

Second, some of the analysis scenarios in this project stem from prior internship experiences of the team members. While all raw data was independently collected and processed by students, a deeper empirical exploration of the results may inadvertently risk disclosing sensitive commercial insights.

In broader terms, our findings have practical implications for financial analysts and researchers who increasingly rely on alternative data sources such as product listings, consumer reviews, and pricing pages. LLM-assisted workflows offer a scalable and resilient framework for unlocking the value embedded in messy web content, thus enhancing the analytical toolkit available to professionals in investment research, competitive benchmarking, and digital market intelligence.

Appendix: Data and Code (accessible via Google Drive links with UCSD accounts)

https://drive.google.com/file/d/1CLAxKQ70JxHgoDZ3dnmDvGI4Of012Z4y/view?usp=drive_link