Don’t Forget This: Augmenting Results with Event-Aware Search

Hugo Sousa*
University of Porto
INESC TEC
Porto, Portugal
hugo.o.sousa@inesctec.pt

Austin R. Ward
Amazon
Seattle, WA, USA
ausrward@amazon.com

Omar Alonso
Amazon
Palo Alto, CA, USA
omralon@amazon.com

Abstract

Events like Valentine’s Day and Christmas can influence user intent when interacting with search engines. For example, a user searching for `gift` around Valentine’s Day is likely to be looking for Valentine’s-themed options, whereas the same query close to Christmas would more likely suggest an interest in Holiday-themed gifts. These shifts in user intent, driven by temporal factors, are often implicit but important to determine the relevance of search results. In this demo, we explore how incorporating temporal awareness can enhance search relevance in an e-commerce setting. We constructed a database of 2K events and, using historical purchase data, developed a temporal model that estimates each event’s importance on a specific date. The most relevant events on the date the query was issued are then used to enrich search results with event-specific items. Our demo illustrates how this approach enables a search system to better adapt to temporal nuances, ultimately delivering more contextually relevant products.

CCS Concepts

• **Information systems → Information retrieval**; **Electronic commerce**.

Keywords

Event-Aware Search, Temporal Modeling, E-Commerce

ACM Reference Format:
Hugo Sousa, Austin R. Ward, and Omar Alonso. 2025. Don’t Forget This: Augmenting Results with Event-Aware Search. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM ’25), March 10–14, 2025, Hannover, Germany.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3701551.3704119

1 Introduction

A search engine is the primary tool for helping users find relevant information, products, or services based on their queries. User queries often carry implicit meaning shaped by various factors, such as temporal information. For instance, as Valentine’s Day approaches, users may expect search results related to gifts or romantic experiences, even without directly referencing the event. Despite the impact of temporal context on search [1, 3], event-driven search remains an understudied area.

In this paper we demo a search system that takes into consideration the importance of upcoming events. We start by building a database of 2K events that we use to model the daily importance of each event based on historical data. Events with validated temporal models are then incorporated into a search system that uses the temporal models to determine which events are relevant on the date the query was issued. The number of event-related items displayed in search results is proportional to the event’s importance as estimated by the temporal model. This ensures that the search results are relevant to the temporal context of the query.

In prior work [9], a language model was used to generate events for search. However, the language model world knowledge of events was found to be limited, leading to the proposal of the structured approach to extract event information described in this manuscript. Another relevant work is EventWiki [2], where the authors extracted 21K events from Wikipedia. However, EventWiki primarily focuses on historical events, whereas our research centers on recurring events.

Recent studies have emphasized the importance of seasonality in search relevance, particularly in e-commerce settings. The work by [10] found that 39% of queries in e-commerce are seasonally sensitive, and accounting for temporal patterns in search rankings significantly improved user engagement metrics. Other researchers have implicitly modeled the temporal importance of events by training a next-item recommendation system [8]. The devised model is named Occasion-aware Recommender and the authors found to surpass the state-of-the-art model in two e-commerce benchmarks. Our work extends this line of research by explicitly augmenting the search results with event information.

2 Event-Aware Search

We’re interested in using event information to enhance search results. Our proposed system consists of three main components. An **Event Database**, which makes the system aware of the breadth of events available in a typical year. **Temporal Modeling** is used to assess the importance of each event on a given day of a calendar year. Finally, **Event-based Retrieval**, uses events and their temporal models to retrieve relevant results.

2.1 Event Database

We use Wikidata [7] and the Time and Date Website [6] as input seed to build the database. The two sources are then aggregated and filtered to produce the final database with 2K events.

Event Sources. Wikidata stores information in a triplet SPO format, which can be queried using SPARQL. For this project, we extract all events that have the property *day in year for periodic occurrence* (id `P837`) defined. This property links recurring events to

*Work done during internship at Amazon.

WSDM ’25, March 10–14, 2025, Hannover, Germany

ACM ISBN 979-8-4007-1329-3/25/03
https://doi.org/10.1145/3701551.3704119

a specific date. For our demo, we limit our scope to events celebrated in the United States or global events. This is achieved by filtering for events where the property *country* (id: `P17`) is set to "United States of America" (id: `Q30`) or is undefined.

We parse Wikidata associated dates as follows. Dates can take various forms, such as a specific day (e.g., "October 27"), a month (e.g., "June"), another event (e.g., "Easter Sunday"), or a relative date (e.g., "fourth Thursday in November"). To resolve these into specific calendar dates, we developed a parser that converts the canonical date descriptions into a list of valid dates from the year 2000 onwards. After applying the parser, 9% of the events were removed due to either invalid date values (e.g., "Not Defined") or unsupported date formats (e.g., dates referencing the lunar calendar or complex relative dates like "the last Sunday before December 25$^{\text{th}}$"). For all these events, we added the Wikipedia page text and summary when available.

For the Time and Date website, we focus on the "Fun Holidays" and "US Holidays" categories. For Fun Holidays, we extract the event name, date, and page content. Since these holidays occur on the same date every year, expanding the data across multiple years is straightforward. For US Holidays, the scrapping process is similar, though additional care is required for holidays which date vary by year (e.g., "Easter Sunday"). In these cases, we scrape the specific date for each year from 2000 onward.

***Event Aggregation & Filtering*.** Because events from Wikidata and Time and Date can overlap, event aggregation from the two sources needs attention. For instance, the same event might appear under different names (e.g., "Saint Patrick's Day" and "St. Patrick's Day"). To address this, we use a language model to create a canonical name for each event. Specifically, we prompt Anthropic Claude 3.5 Sonnet to generate a concise event name based on the extracted event names and the available information that we captured. The generated canonical name is used to aggregate the information from the two sources.

Finally, we implemented an adult content filter to exclude events that could be offensive, insensitive, or inappropriate. Since we generate an image for each event on our demo, this filtering is applied directly to the generated event images by using an NSFW filter.

2.2 Temporal Modeling

Our temporal model estimates the relative importance of an event by measuring the volume of event-related queries issued on any given day. Algorithm 1 details our approach.

We first build the purchase dataset using historical data where we store date, query, purchased item. Then, for each event, we define a timeframe of 90 days before the event start and 60 days after the event end to identify relevant products and queries for that event (lines 2-3). The relevant products are found by doing semantic matching between the event name and the products in that timeframe (lines 4-6). From these relevant products we then find all the queries that lead to the purchase of those products, defining those as the relevant queries for the event (line 7). We then track the daily frequency of the relevant queries to produce the baseline temporal signal (line 8).

Since some of the queries deemed relevant might also occur throughout the year, this signal contains some baseline frequency.

Algorithm 1 Event Temporal Modeling

Require: E: Event with name and start date; D: Purchase dataset with date (d), query (q), and product (p); Embed: Function to embed text; Distance: Distance function; th: Distance threshold for relevance

Ensure: M: Temporal model of the event (percentage of relevant queries)

1: **procedure** TemporalModel(E, D)
2: $T \leftarrow [E.\text{start} - 90, E.\text{end} + 60]$ ▷ Timeframe
3: $D_T \leftarrow \{(d, q, p) \in D : d \in T\}$ ▷ Dataset filtered to timeframe
4: $E_{emb} \leftarrow \text{Embed}(E.\text{name})$ ▷ Embed event name
5: $P_{emb} \leftarrow \{\text{Embed}(p) : p \in D_T.\text{products}\}$ ▷ Embed all products
6: $P_r \leftarrow \{p : \text{Distance}(E_{emb}, P_{emb}[p]) \leq th\}$ ▷ Relevant products
7: $Q_r \leftarrow \{q : \exists p \in P_{rel}, (q, p) \in D_T\}$ ▷ Relevant queries
8: $S \leftarrow \text{DailyFrequency}(Q_{rel}, D_T)$ ▷ Baseline signal
9: $ed \leftarrow \text{Mean}(S) + \text{StandardDeviation}(S)$ ▷ Event duration estimate
10: $S_{MA} \leftarrow \text{MovingAverage}(S, 4 \times ed)$
11: $F_{MA} \leftarrow \text{MovingAverage}(S, 3)$
12: $t_{date} \leftarrow \text{Takeoff}(F_{MA}, S_{MA}, ed)$
13: $f_{date} \leftarrow \text{Dropoff}(F_{MA}, S_{MA}, ed)$
14: $S_e \leftarrow \{(d, c) \in S : t_{date} \leq d \leq f_{date}\}$ ▷ Event signal between takeoff and dropoff
15: $M \leftarrow \mathit{SumNormalize}(S_e)$
16: **return** M
17: **end procedure**

To determine when an event gains momentum (takeoff) and when it diminishes (dropoff), we compute two moving averages: a slow-moving average which captures stable patterns, and a fast-moving average that reacts quickly to transient fluctuations (lines 9-11). The slow-moving average is set to four times the event's estimated duration – estimated with the number of days the baseline temporal signal remains above the mean plus one standard deviation – while the fast-moving average is set to three days. The takeoff date is defined as the first day when the fast-moving average stays above the slow-moving average for more than half the estimated event duration, and the dropoff point is when it remains below for the same period (lines 12-13). The temporal model of an event is then defined by the normalized sum of the signal between the takeoff and dropoff points (lines 14-15). Figure 1 illustrates this process for Valentine's Day, showing the baseline signal along with the fast and slow moving averages, and the takeoff and dropoff dates.

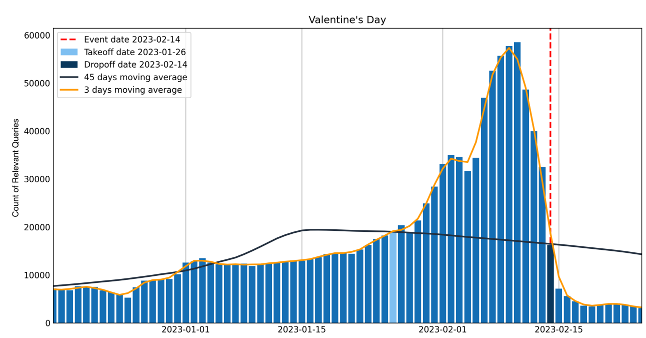

Figure 1: Baseline temporal signal for Valentine's Day with the fast and slow moving averages, takeoff and dropoff dates.

Regarding the practical implementation details, we collected purchase data for over 1 year. The semantic matching is made with the `all-MiniLM-L12-v2` as the embedding model [5], for the distance metric we used the L2 distance, and 1.2 as the distance threshold for relevance (*th* in Algorithm 1). These values were set empirically and could be refined by applying the temporal models to a downstream task that has a concrete evaluation benchmark.

Since not all events lead to observable user purchase behavior, for this demo, we manually filter out events where takeoff and dropoff dates did not clearly distinguish the event from the slow-moving average. This process resulted in a set of 16 very popular events (e.g., "Valentine's Day", "Saint Patrick's Day", "Christmas").

2.3 Event-based Retrieval

The devised temporal models are used to contextualize the search results with upcoming events. To illustrate the search system implemented in our demo, we use the query `gifts` as if it was issued on February 7 as a running example. Figure 2 depicts the steps for this example, which are detailed in the following paragraphs.

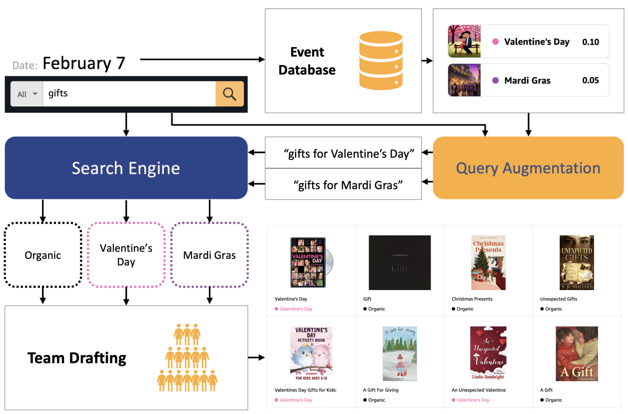

Figure 2: Graphical representation of the search system processing the query `gifts` issued on February 7.

When processing a query, the system first references the event database to find events whose temporal models include the query date. These events are deemed relevant for that day and are returned with the corresponding weight from the temporal model. In our example, on February 7, two events are considered relevant: "Valentine's Day" and "Mardi Gras," with relative importance of 0.1 and 0.05, respectively [1].

Next, query expansion is performed by adding the event name to the original query. In our example, the query `gifts` is expanded to `gifts for Valentine's Day` and `gifts for Mardi Gras`. We then retrieve search results for both the original and event-expanded queries yielding organic and event related results. In our demo, the search engine is implemented with dense retrieval, comparing the query with product names in a database of two million items using cosine similarity as the distance metric and `all-MiniLM-L12-v2` as the embedding model [5].

Given the different result sets, we employ a *team drafting* approach to merge them and generate the final result list [4]. The allocation of slots for each event's results is proportional to the event's importance on the given date, while the remaining slots are filled with organic results. Specifically, if the total number of results is k, the number of slots for an event is set to $2 \times E_i \times k$, where E_i represents the event's importance that day.

In our running example, Valentine's Day has an event importance of 0.1. For $k = 32$, the number of slots allocated to Valentine's Day-related results is $2 \times 0.1 \times 32 = 6.4$, which is rounded down to 6 slots. Similarly, Mardi Gras, with an event importance of 0.05, receives $2 \times 0.05 \times 32 = 3.2$, rounded down to 3 slots. The remaining 23 slots are populated with organic results.

After combining the results according to the slot allocation, the final list is presented to the user.

[1]These values are fictional and provided solely for illustrative purposes.

3 Demonstration

Our demo is designed to showcase the three key elements of our research. The events from our database are presented through **Event Cards**, accessible via a calendar. Users can interact with the event temporal models in the **Event Page**. Finally, the **Search Page** shows the event-based retrieval system in action.

3.1 Event Card

By default, when the user opens the demo website, the page displays an event card for each event occurring on that day. As shown in Figure 3, each card includes an image, name, and a brief description. The short descriptions were generated using Claude 3.5 Sonnet which was prompted to create concise summaries from the information we had for each event. The images accompanying the events were produced using a diffusion model (Stability AI SDXL) which was prompted with the event name and description to generate images. To search for events on different days, users can interact with a calendar; by clicking on a specific date, the page displays event cards for the events happening on that day. This functionality is showed with the "Valentine's Day" event in Figure 3.

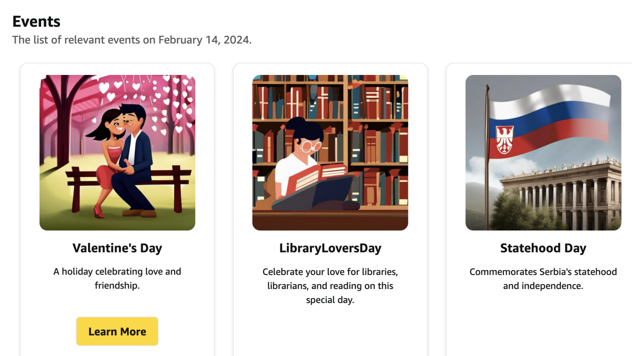

Figure 3: Three event cards for February 14, 2024.

3.2 Event Page

Once on the Event Page, shown in Figure 4, the user can see and interact with the temporal model devised for that event. The temporal distribution is displayed in the top-right corner, alongside a slider that spans from the takeoff date to the dropoff date. By adjusting the temporal slider, the page dynamically updates the "Best Sellers" and "Top Queries" sections for the selected date.

The "Best Sellers" for each day are determined by identifying relevant products for the event during the temporal model's computation. For the "Top Queries," we first perform semantic clustering of the relevant queries for that day. Queries are clustered based on cosine similarity between their embeddings, computed with the `all-MiniLM-L12-v2` model [5]. Queries with a cosine similarity above 0.7 are grouped into the same cluster. The query representing each cluster on the Event Page is chosen based on its importance, which is defined by the number of queries it is connected to within the cluster. If multiple queries share the highest importance, the query with the shortest character length is selected to be displayed.

This interface allows us to illustrate products and queries that may be highly relevant to an event on any given date. For example, in the case of "Halloween," products like movies are more popular well in advance of the event, while costumes and makeup become more prominent as the event draws near (Figure 4).

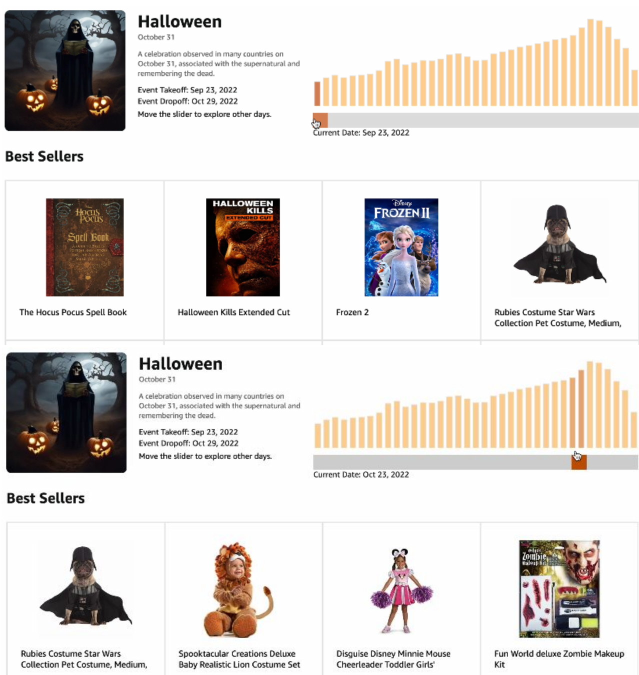

Figure 4: Event Page for Halloween. Top: the date is set for Sep 23rd, showing more movies. Bottom: the date is set for Oct 25th, showing more costumes.

3.3 Event-Aware Search Page

The Event-Aware Search Page shows the functionality of the retrieval system described in Section 2.3. This page includes a temporal slider that spans from January 1 to December 31, allowing users to input a query in the search box and adjust the slider to observe how the search results change based on the query's issuance date. As the slider moves, the events being taken into consideration for that search appear in small cards between the slider and the search results, as illustrated in Figure 5.

The search engine is configured to display n products, each showing the product image, name, and result source. The source can either be organic, indicating the result comes from the original query's search results, or the event name, indicating the product was retrieved from the augmented query with that event name.

Figure 5 presents the first four results for the query `gifts`, as if it were issued on February 7. On this date, two events are deemed relevant, namely: "Valentine's Day" and "Mardi Gras". And so search results related to those two events are shown together with organic search results as described in Section 2.3.

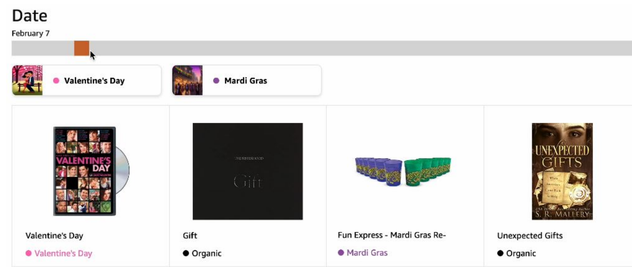

Figure 5: Event-Aware Search Page when queried with `gifts` with the temporal slider (grey bar with dark orange square at the top) set to February 7.

4 Conclusion & Future Work

In this paper, we demonstrate how events can be used as a signal to enhance the relevance of search results. However, a user study is needed to assess the impact of the proposed system. In addition, there are still several areas within this research that remain unexplored: 1) assessing if a given query is directly related to an event through a classifier; 2) comprehensive reranking of all the drafted results; 3) incorporating single-occurrence or trending events (e.g., sports matches, concerts, stock market changes, fashion trends); and 4) the impact of personalization (e.g., shopper's preferences, location). These particular areas will be a focus as we continue researching this area.

References

[1] Ricardo Campos, Gaël Dias, Alípio M Jorge, and Adam Jatowt. 2014. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)* 47, 2 (2014), 1–41.

[2] Tao Ge, Lei Cui, Baobao Chang, Zhifang Sui, Furu Wei, and Ming Zhou. 2018. Eventwiki: a knowledge base of major events. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

[3] Nattiya Kanhabua and Kjetil Nørvåg. 2012. Learning to rank search results for time-sensitive queries. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*.

[4] Eugene Kharitonov, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. 2015. Generalized Team Draft Interleaving. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*.

[5] sentence-transformers. 2024. all-MiniLM-L12-v2. https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2. Accessed: 2024-09-24.

[6] Time and Date AS. 2024. Time and Date. https://www.timeanddate.com/. Accessed: 2024-09-05.

[7] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[8] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to Shop for Valentine's Day: Shopping Occasions and Sequential Recommendation in E-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*.

[9] Austin R Ward and Omar Alonso. 2024. Empowering shoppers with event-focused search. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*. 5294–5298.

[10] Haode Yang, Parth Gupta, Roberto Fernández Galán, Dan Bu, and Dongmei Jia. 2021. Seasonal relevance in e-commerce search. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*. 4293–4301.