



AUTOCOMPLETE FUNCTIONALITY IN SEARCH ENGINES

DEBADRITA ROY

BCSE-IV

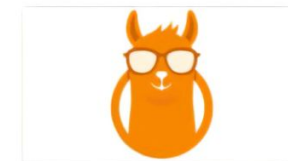
001910501025



INTRODUCTION **SEARCH ENGINES**

WHAT IS A SEARCH ENGINE?

- ❏ A search engine is a software system designed to carry out web searches.
- ❏ They search the World Wide Web in a systematic way for particular information specified in a web search query.
- ❏ The search results are generally presented in a line of results often called search engine results pages (SERPs). The information may be a mix of links to web pages, images, videos, infographics, articles, research papers, and other types of files.



HISTORY OF SEARCH ENGINES

- ❑ The first well documented search engine that searched content files was Archie (debuted on 10 September 1990). The program downloaded the directory listings of all the files located on public anonymous FTP (File Transfer Protocol) sites, creating a searchable database of file names; however, it did not index the contents of these sites since the amount of data was so limited it could be readily searched manually.
- ❑ JumpStation (created in December 1993 by Jonathon Fletcher) used a web robot to find web pages and to build its index, and used a web form as the interface to its query program. Because of the limited resources available on the platform it ran on, its indexing and hence searching were limited to the titles and headings found in the web pages the crawler encountered.
- ❑ WebCrawler, which came out in 1994, allowed users to search for any word in any webpage, which has become the standard for all major search engines since. It was also the search engine that was widely known by the public.

HISTORY OF SEARCH ENGINES (CONTD)

- ❑ The first popular search engine on the Web was Yahoo! Search. In 1995, a search function was added, for searching Yahoo! Directory. It became one of the most popular ways for people to find web pages of interest, but its search function operated on its web directory, rather than its full-text copies of web pages.
- ❑ In 1996, Robin Li developed the RankDex site-scoring algorithm for search engines results page ranking.
- ❑ Around 2000, Google's search engine rose to prominence. The company achieved better results for many searches with an algorithm called PageRank which ranked web pages based on the number and PageRank of other websites and pages that link there, on the premise that good or desirable pages are linked to more than others.
- ❑ As of 2019, active search engine crawlers include those of Google, Petal, Sogou, Baidu, Bing, Gigablast, Mojeek, DuckDuckGo and Yandex.



search engine



All

Images

Books

News

Videos

More

Tools

About 1,23,00,00,000 results (0.49 seconds)

<https://www.searchenginejournal.com> › Tools

20 Great Search Engines You Can Use Instead of Google

23-Sept-2021 — 20 Great **Search Engines** You Can Use Instead of Google · 1. Bing · 2. Yandex · 3. CC Search · 4. Swisscows · 5. DuckDuckGo · 6. StartPage · 7. Search ...
[Meet the 7 Most Popular...](#) · [The 10 Best Video Search...](#) · [Image Search](#)

People also ask

What are the 5 top search engines?



What is search engine and example?



What is search engines?



What are the 4 types of search engines?




[Feedback](#)

<https://en.wikipedia.org> › wiki › Search_engine

Search engine - Wikipedia

A **search engine** is a software system designed to carry out web searches. They search the World Wide Web in a systematic way for particular information ...



Search engine

Software type

A search engine is a software system designed to carry out web searches. They search the World Wide Web in a systematic way for particular information specified in a textual web search query. The search results are generally presented in a line of results, often referred to as search engine results pages.

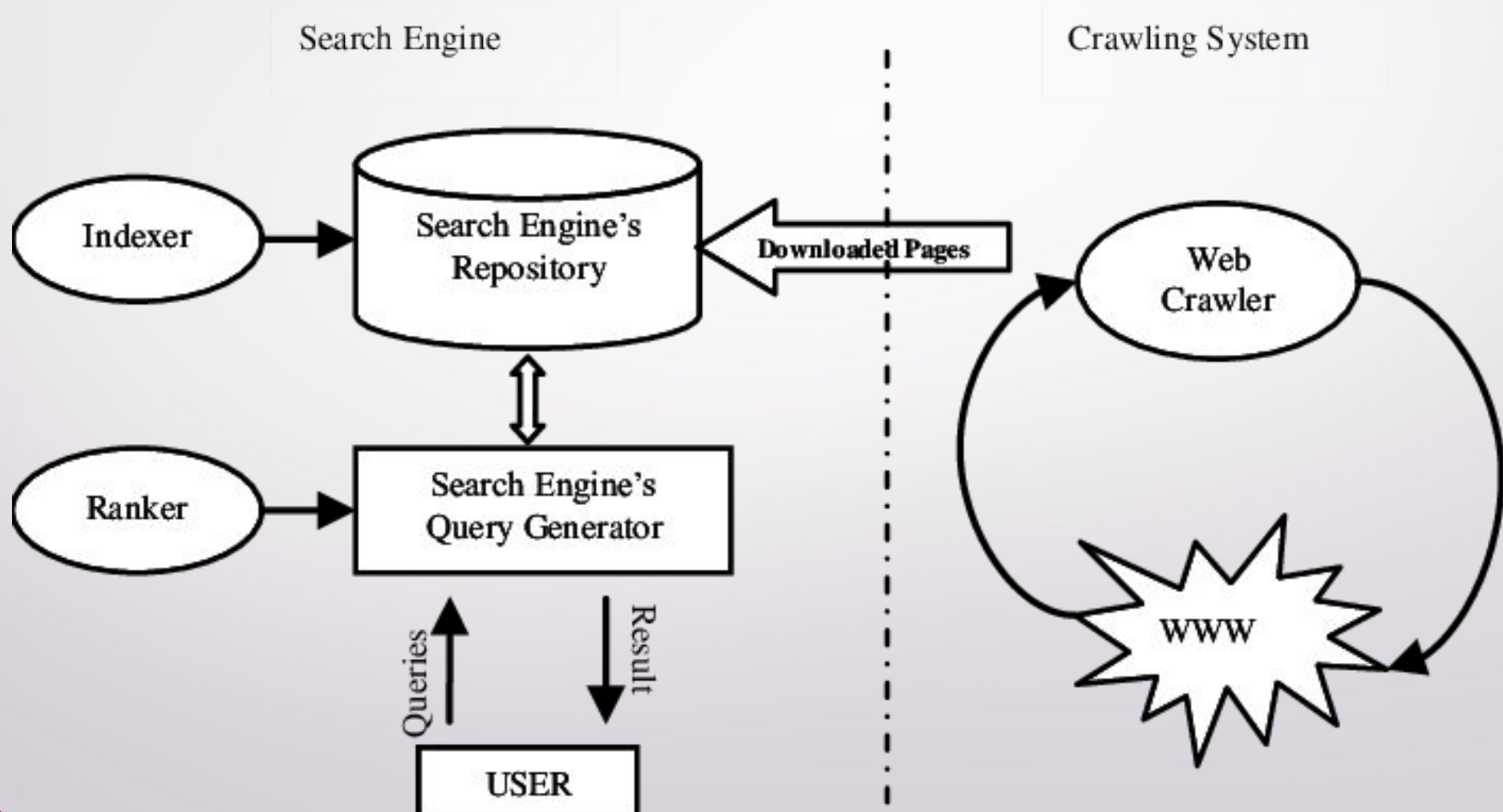
[Wikipedia](#)

[Feedback](#)

SEARCH ENGINE COMPONENTS

1. **Web Crawler:** mainly a software component that traverses on the web, then downloads and collects all the information over the Internet
2. **Database:** the place where all the web information is stored
3. **Search Interface:** interface between the user and the database which helps users to search for queries using the database
4. **Ranking Algorithm:** to rank web pages according to the some features like location and frequency, link analysis and click-through measurement

WORKING OF A SEARCH ENGINE





AUTOCOMPLETE FUNCTIONALITY **BACKGROUND**

AUTOCOMPLETE: WHAT IS IT?

- ❏ Autocomplete, or word completion, is a feature in which an application predicts the rest of a word a user is typing.
- ❏ It attempts to anticipate search terms based on the behavior, previous searches, geolocation, and other attributes of the end user, as well as trending searches across all user sessions, and displays the possible suggestions in or under the search field.
- ❏ Many autocomplete algorithms learn new words after the user has written them a few times, and can suggest alternatives based on the learned habits of the individual user.
- ❏ Autocomplete feature is seen in almost every app and website today and is especially helpful when one is typing on a mobile, as we do not need to type the entire phrase/query on the small screen.

AUTOCOMPLETE IN SEARCH ENGINES

- ❏ Autocomplete user interface features provide users with suggested queries or results as they type their query in the search box (also commonly called autosuggest or incremental search).
- ❏ When the writer writes the first letter or letters of a word, the program predicts one or more possible words as choices. If the word they intend to write is included in the list they can select it. If the word is not predicted, the writer must enter the next letter of the word. At this time, the word choice(s) is altered. When the word that the user wants appears it is selected, and the word is inserted into the text.
- ❏ The challenge is to search large indices or popular query lists in under a few milliseconds so that the user can see increasingly accurate results pop up while typing.



HOW IT CAME INTO BEING: THE HUMBLE ORIGINS OF AUTOCOMPLETE

Google search was the first organization to implement an autocomplete feature. As the story goes, the idea was born on a Google shuttle bus. Kevin Gibbs, a Stanford grad and a former IBM engineer, had joined Google because he liked the shuttle service that the company provided its employees and the flexibility Google used to offer its engineers to spend a fifth of their time working on projects of special interest to them. He wanted to find a way to take advantage of the technological advances of the time -- big data, JavaScript, the broadening consumer use of high-speed Internet -- to make web navigation more efficient. So Gibbs decided to spend his own 20 percent time working on a URL predictor. As a user typed a URL into a browser, Gibbs's system would analyze Google's enormous corpus of web content, and then autocomplete the options that remained.



HOW IT CAME INTO BEING: AUTOCOMPLETE AS WE KNOW IT

When Gibbs showed his new feature to his coworkers, one of them said, "That's cool, what if you did it for search?"

From there Google's internal infrastructure took over. Google's heads of search, including Jeff Dean and Rob Pike, began promoting Gibbs's work within the company. Marissa Mayer helped name the service, favoring "Google Suggest" over Gibbs's name ("Google Complete"). The feature launched on December 10, 2004. Google Suggest would remain an opt-in feature for four more years, until, in 2008, Google made autocomplete the default search mode on both Google.com and the company's mobile apps, maps, and browsers. In 2010, Google expanded the feature to Google Instant. Autocomplete has now become an expected feature, and has been implemented by many organizations, big and small.



ORIGINAL BLOG POST LAUNCHING AUTOCOMPLETE

I've got a suggestion

December 10, 2004

Today we launched [Google Suggest](#), a new Labs project that provides you with search suggestions, in real time, while you type. We've found that Google Suggest not only makes it easier to type in your favorite searches (let's face it -- we're all a little lazy), but also gives you a playground to explore what others are searching about, and learn about things you haven't dreamt of. Go ahead, [give it a spin](#).

The project stemmed from an idea I had a few months ago, and since then I've been working on it in my 20% time, which is a program where Google allows their employees to devote 20% of their working hours to any project they choose. What's really amazed me about this project is how in a matter of months, working on my own, I was able to go from a lunch table conversation to launching a new service. In my opinion, this is one of the things that really makes Google a great place; that the company's systems, resources and, most important, people are all aligned to make it as easy as possible to take an idea and turn it into something cool.

Plus, we have Segways.

Kevin Gibbs

Software Engineer

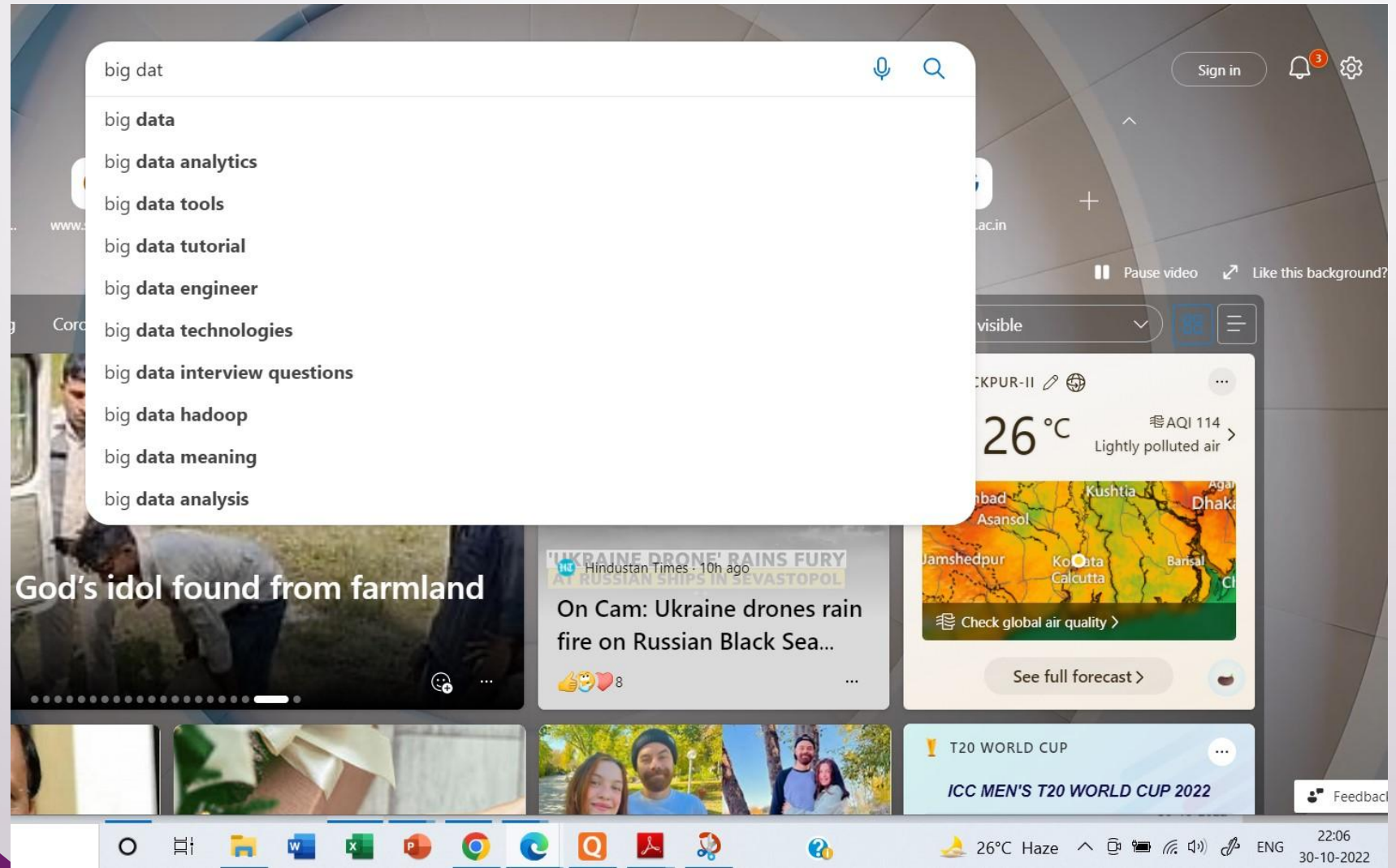
BIASES AND PERSONALIZATIONS IN AUTOCOMPLETE

The suggestions given by the autocomplete feature depend on

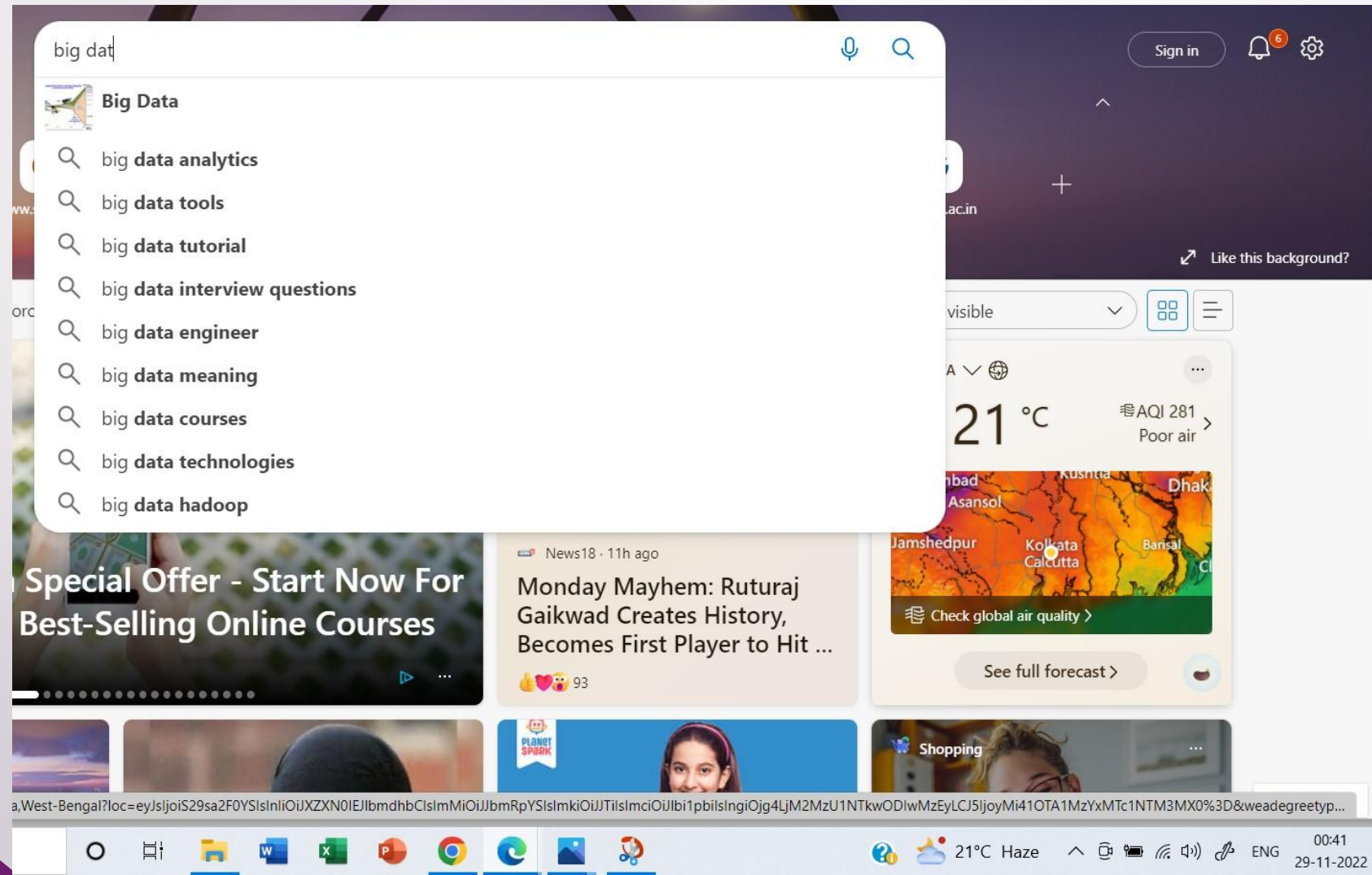
- the previous search history of the user(s),
- time (suggestions change over a period of time, either the order of the suggestions changes or new suggestions come up or past suggestions are discarded), and
- the search engine used (as search engines use different variations in algorithms to implement the autocomplete feature for web query search).

The following slides show a series of experiments run where “big dat”, etc. is searched and the autocomplete suggestions are noted.

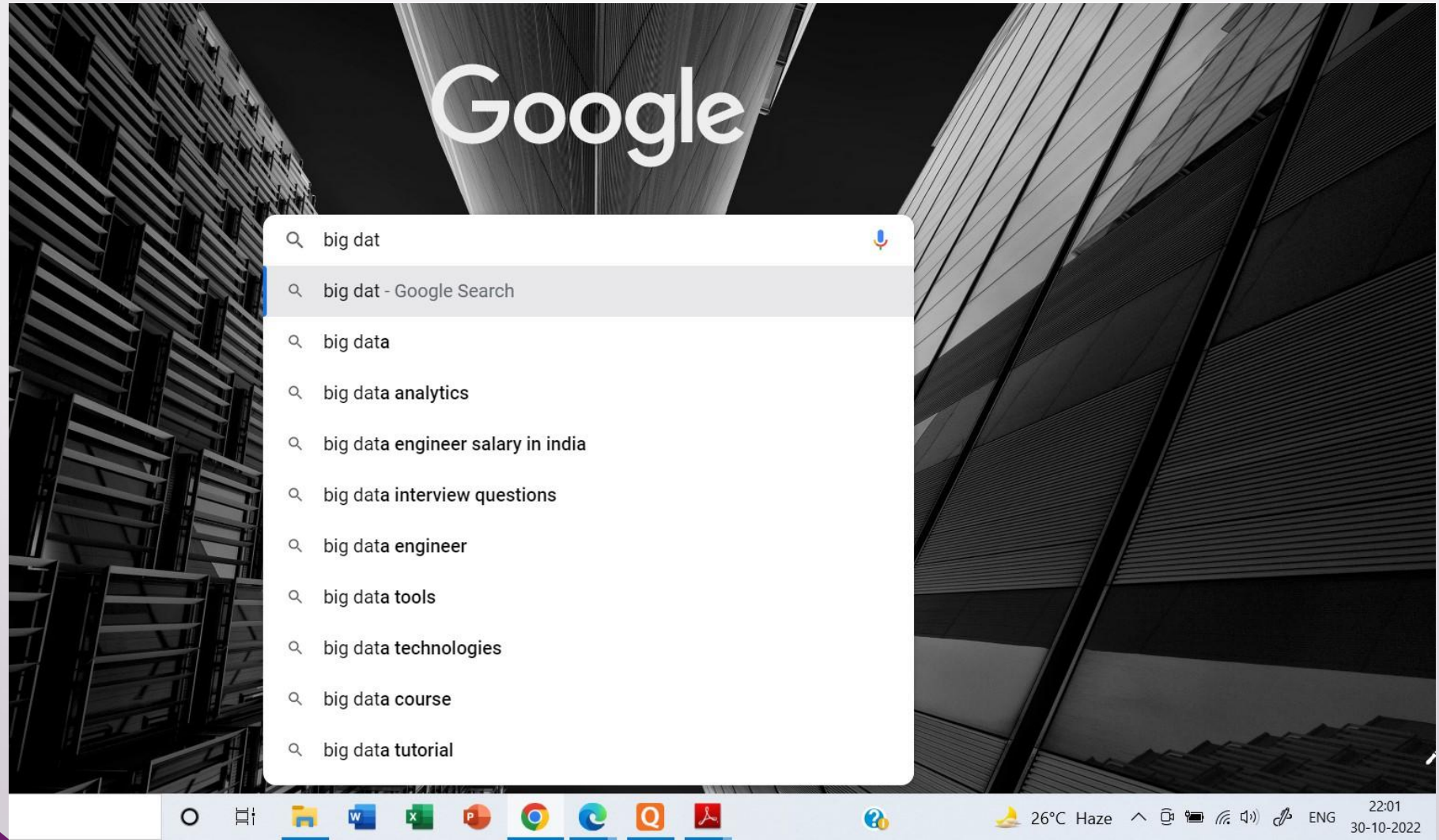
SEARCHED ON MICROSOFT EDGE AT DIFFERENT TIMES (30th October 2022)



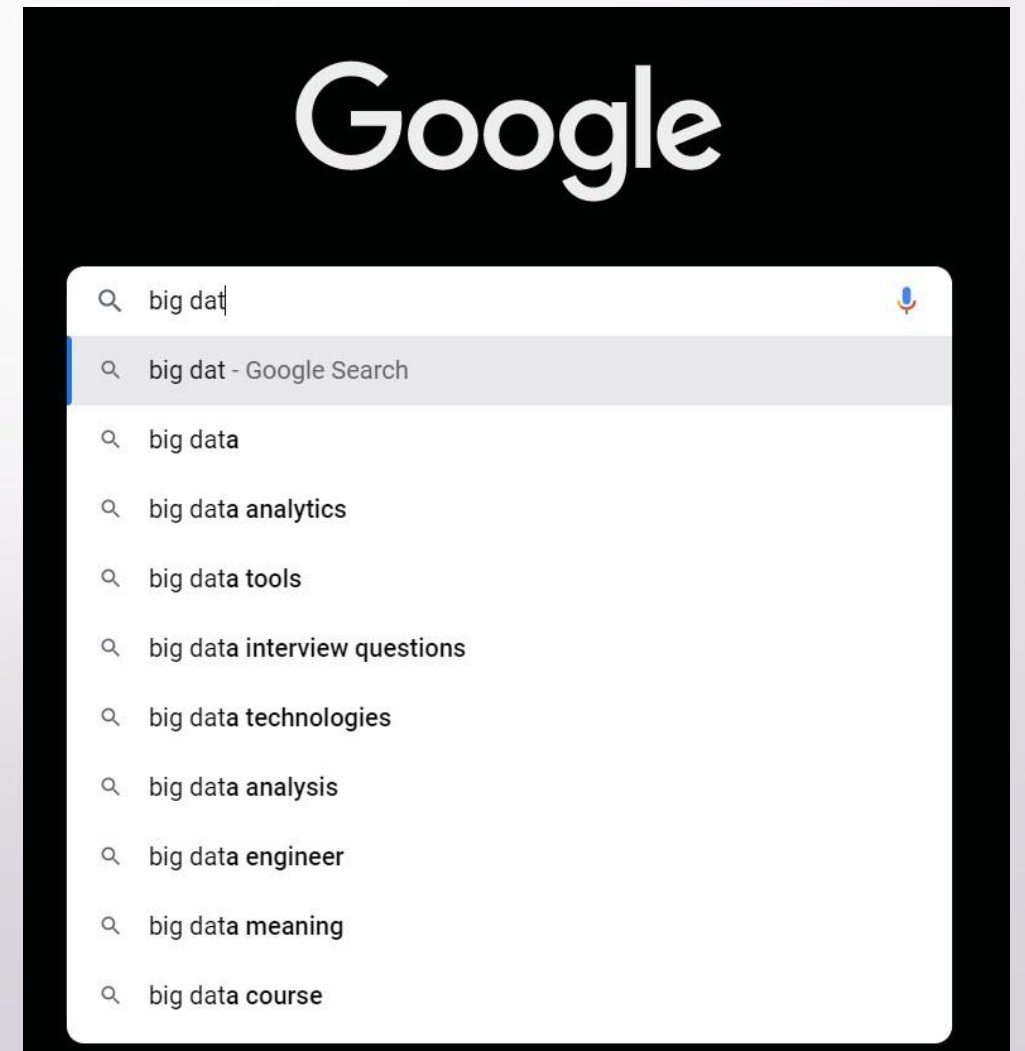
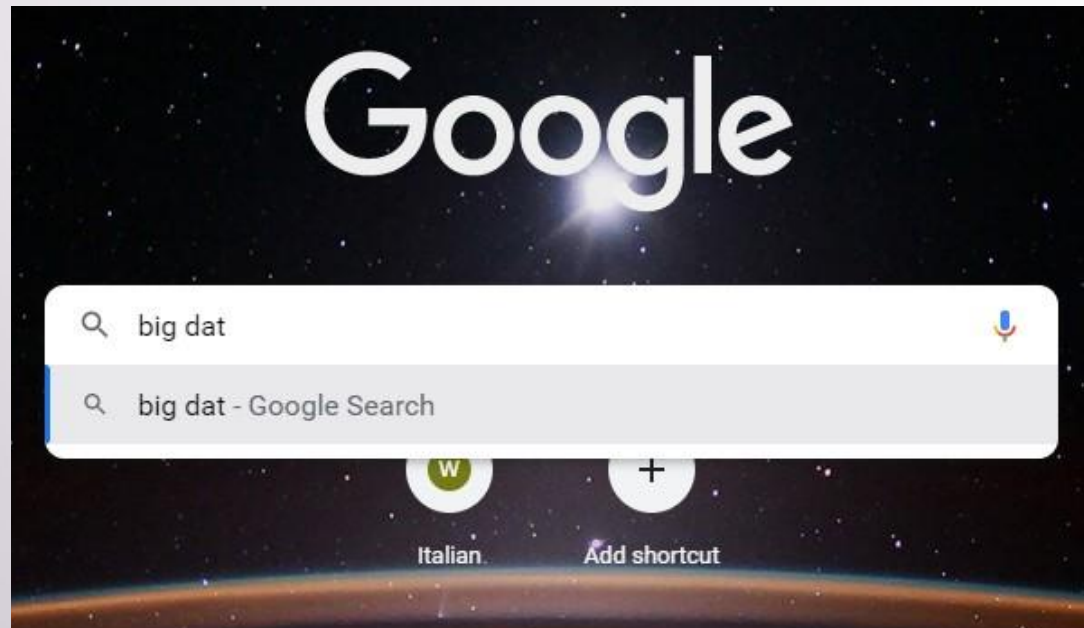
SEARCHED ON MICROSOFT EDGE AT DIFFERENT TIMES (29th November 2022)



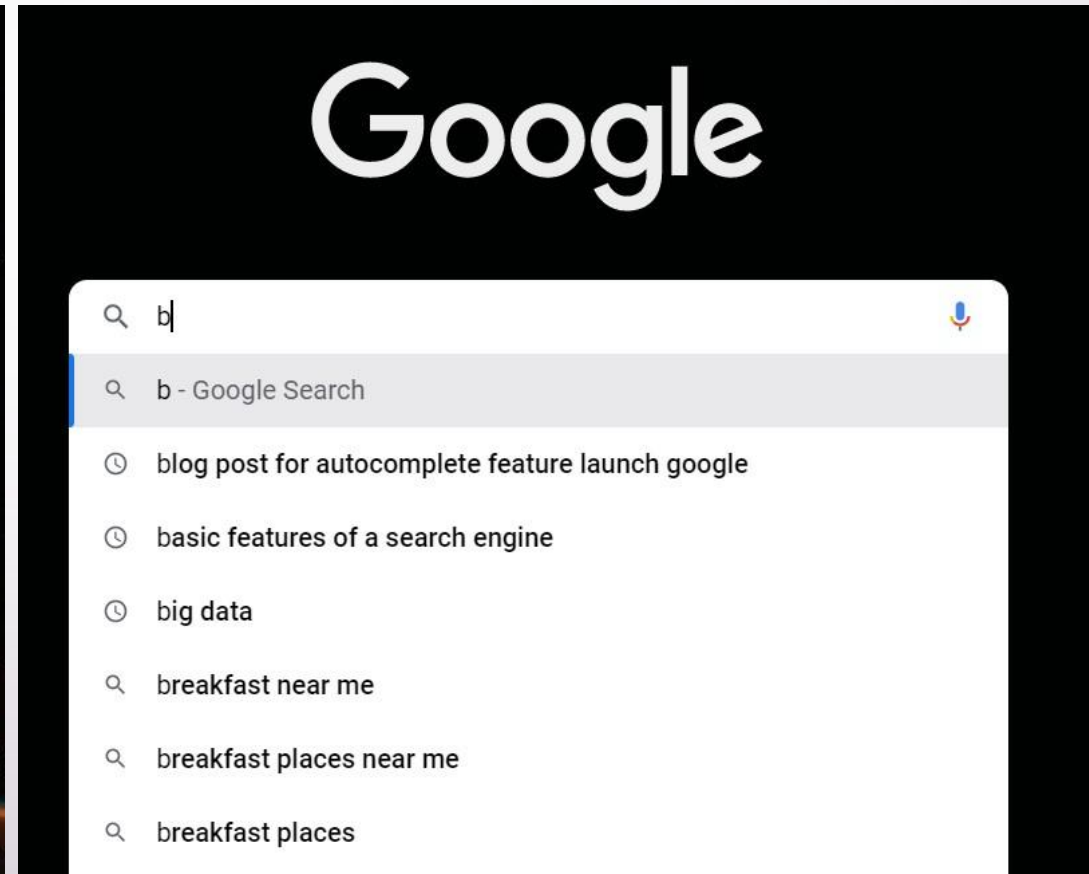
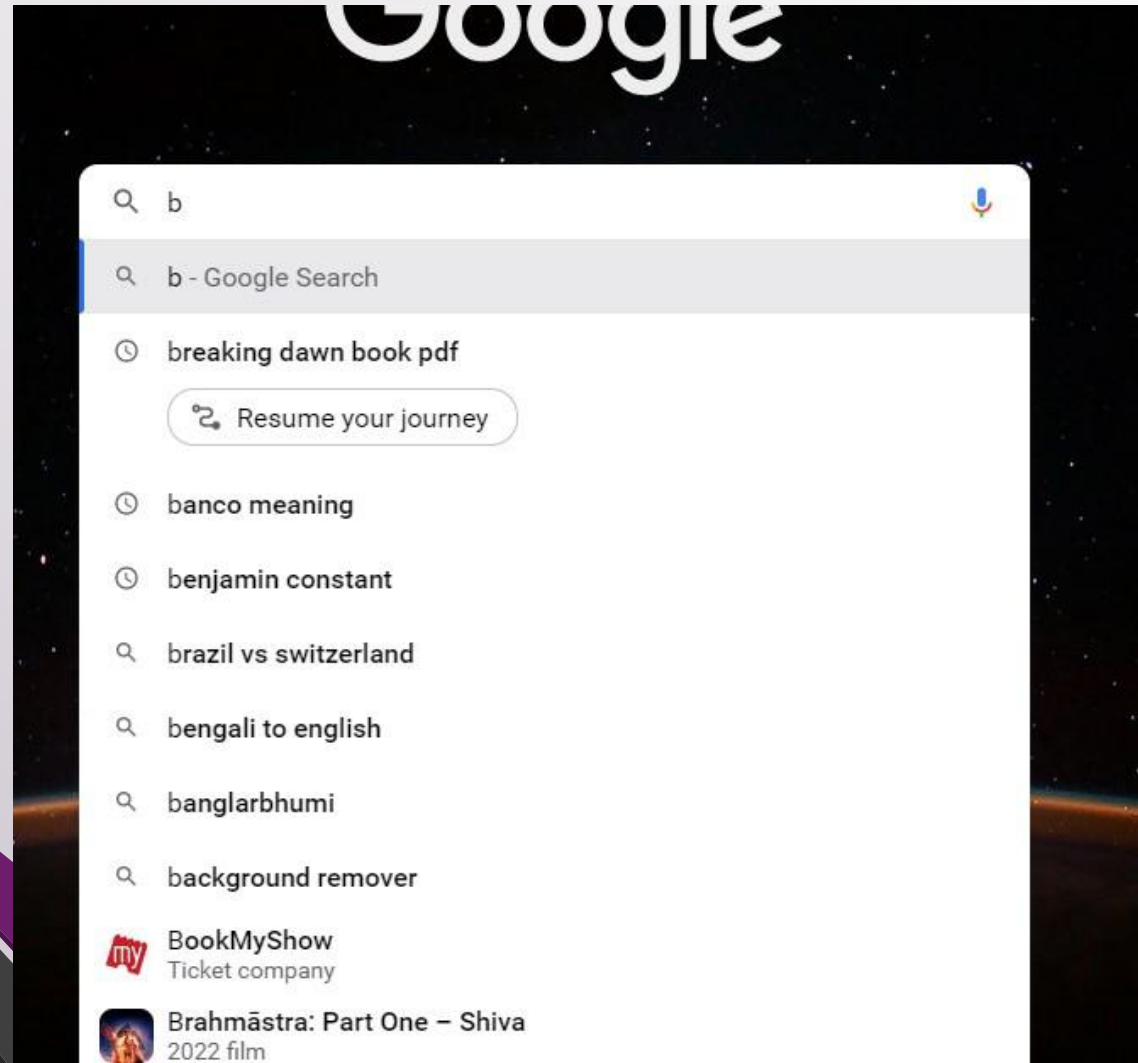
SEARCHED ON GOOGLE CHROME AT THE SAME TIME (30th October 2022)




SEARCHED ON GOOGLE CHROME BY DIFFERENT USERS (29th November 2022)



SEARCHED ON GOOGLE CHROME BY DIFFERENT USERS (29th November 2022)





AUTOCOMPLETE FUNCTIONALITY
WORKING

AUTOCOMPLETE: HOW DOES IT WORK?

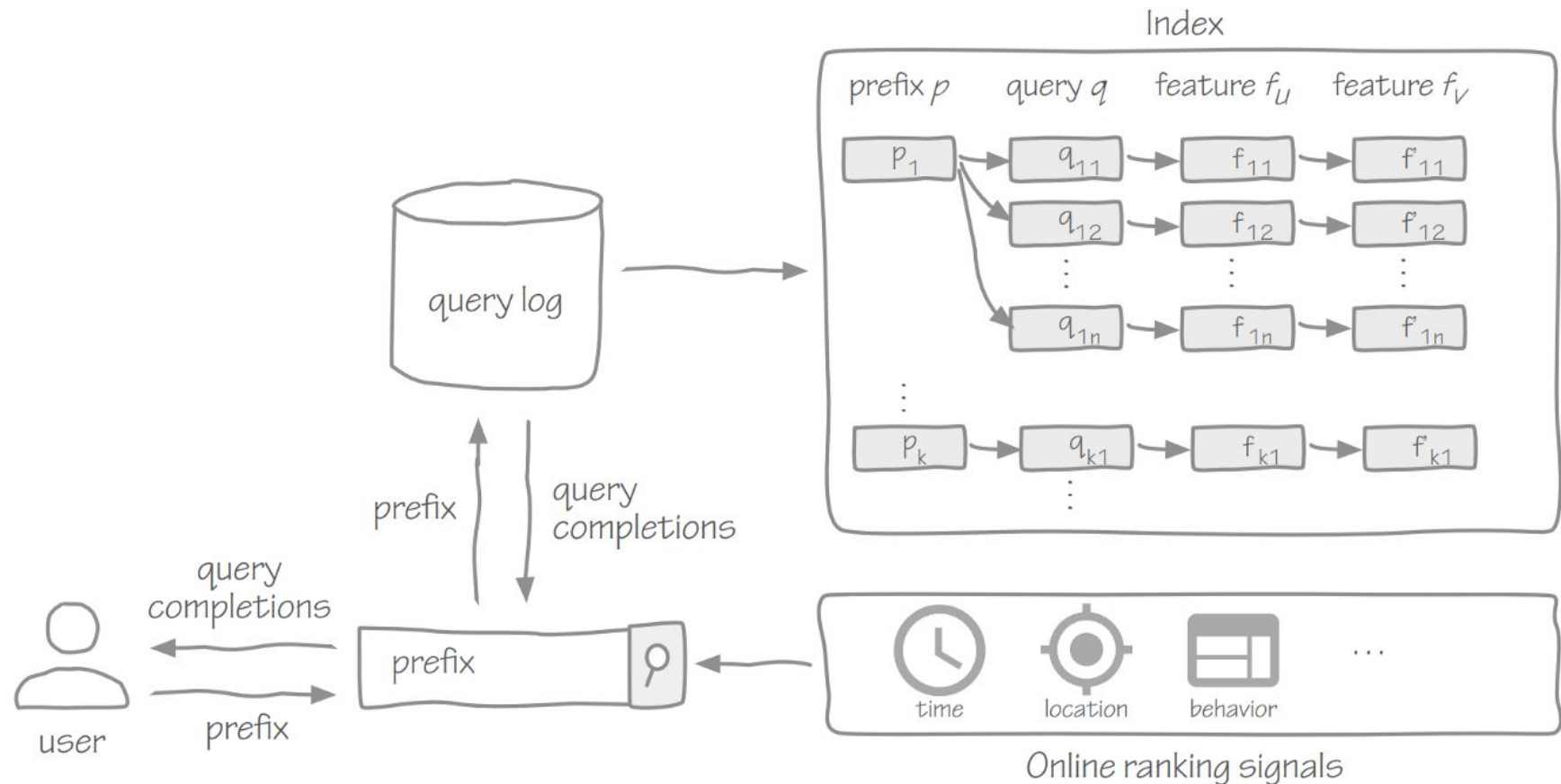
- ❑ It analyzes a search query as it's being typed and provides a set of suggestions, like spelling corrections or a completed search query, in a drop-down menu before the user has even finished typing.
- ❑ Multiple machine learning and natural language processing algorithms and models are used to generate matches, starting with a simple string in order to identify, match, and predict the outcome of the unfinished search query.
- ❑ Word prediction uses language modeling, where within a set vocabulary the words are most likely to occur are calculated. Along with language modeling, basic word prediction is often coupled with a frequency (frequency combined with recency) model, where words the user has used recently and frequently are more likely to be predicted.



FACTORS USED IN QUERY PREDICTION

1. The frequency by which the keywords are used
2. The time (trending keywords have a priority)
3. The language of the query
4. Search history of the individual user
5. Geographical Location of the user

BASIC AUTOCOMPLETE FRAMEWORK



BIG DATA ANALYTICS IN AUTOCOMPLETE

- ★ The suggested words come from a predefined vocabulary, for example all distinct words in the Oxford English Dictionary. The number of words in the vocabulary can be quite large.
- ★ The vocabulary grows even larger for recognizing multi-word phrases and entity names.
- ★ Autocomplete suggestions also take into consideration the trending queries worldwide and the related search queries of other users. The volume and the variety of this data are formidable.
- ★ Words entered by the individual user are prime candidates for completions in the future. To that end, the entered words are inserted into the vocabulary as well.
- ★ The challenge is to search through the corpus and suggest possible queries (based on some ranking algorithm) to the user while they are typing.

APPROACHES USED IN AUTOCOMPLETE

Existing approaches to rank candidate query completions can be divided into two broad categories:

- 1. Heuristic models:** Aim to compute a score directly by considering different sources for each query autocomplete that indicates how likely it is that the query would be issued. Basically, these aim to compute a probability of submitting a query completion q , given some prefix p , at time t and for user u .
- 2. Learning-based models:** Aim to extract dozens of reasonable features to capture the characteristics of each query completion. The model is trained on prefix-query pairs, which can be associated with labels, “submitted” or “non-submitted”. Eventually learn a ranking function from the extracted features.

HEURISTIC MODELS

- **Time-sensitive query autocompletion (QAC) models**
 - **Most popular completion model (Bar-Yossef and Kraus, 2011):** Query candidates are ranked according to their past popularity, calculated as the query's frequency inside the query log normalized by the total sum of query frequencies. Essentially, this model assumes that the current query popularity distribution will remain the same as that previously observed. Thus, it is not sensitive to very fresh real-time events, prioritising long-term popular queries over recent ones.
 - **Models replacing the query's actual frequency with a predicted frequency or modeling the query frequencies using a time-series** have been proposed as it is assumed that temporal data often influences the queries to be entered.

HEURISTIC MODELS

- **User-centred query autocompletion (QAC) models**
 - **Bar-Yossef and Kraus (2011)** treat the user's preceding queries in the current session as context and propose a context-sensitive query autocompletion algorithm that produces completions most similar to the context queries (cosine similarity measure is used). By doing this, a ranked list can be produced. At the same time, another ranked list can be produced based on query popularity. The final ranked list is then constructed by aggregating them.
 - The search context can also be extracted from a user's long-term search history, e.g., the most frequently submitted queries of a particular user. **Cai et al (2014)** propose to personalise QAC by scoring the completions based on both the recent queries in the current search session and the queries previously issued by the same user, if available.

HEURISTIC MODELS

- **User-centred query autocompletion (QAC) models**
 - **Li et al (2015)** propose a probabilistic model that captures the relationship between a user's sequential behaviours at different keystrokes.
 - **Zhang et al (2015)** study implicit negative feedback from a user's interactions with a search engine, and propose an adaptive model that adapts autocompletion to a user's implicit negative feedback about skipped query completions. This model assumes that top ranked but skipped query completions are not likely to be submitted.

LEARNING-BASED MODELS

- **Learning from time-related characteristics**
 - **Cai and de Rijke (2016)** propose to generate a ranking function by learning both observed and predicted query popularity. The predicted query popularity is generated by considering the recent trend as well as the cyclic behaviour of query popularity. They also extract features from so-called homologous queries of a completion (given a query q , queries which extend q or are a permutation of the terms in q) and weigh their contribution to the score of the candidate query completion.
 - **Chien and Immorlica (2005)**, assume that two queries are semantically related in the sense of temporal correlation if their popularity behaves similarly over time. Pearson's correlation coefficient is used to capture this notion of similarity.
 - **Jiang et al (2014)**, incorporate features extracted from user behaviour in sessions in their model.

LEARNING-BASED MODELS

- **Learning from user interactions: Shokouhi (2013)**
 - ❑ Presented a supervised framework for personalised ranking of query auto completions, where several user-specific and demographic-based features were extracted for learning, e.g. gender and location.
 - ❑ To define short-term history features, the queries previously submitted in the current search session were used.
 - ❑ To define long-term history features, the entire search history of the user was considered.
 - ❑ The similarity features of query completions could be measured by n-gram similarity to the context queries.
 - ❑ Experimental results showed that demographic features such as location were more effective for personalising query auto completions.

LEARNING-BASED MODELS

- **Learning from user interactions: Mitra (2015)**
 - ❏ Built on the insight that search logs contain lots of examples of frequently occurring patterns of user reformulation of queries.
 - ❏ Represent query reformulations as vectors, which is achieved by studying the distributed representation of queries learnt by deep neural network models, like the convolutional latent semantic model.
 - ❏ Based on the query representation learnt by the model, cosine similarity between the vectors corresponding to a query completion and the previous queries in the same session are computed and used as features.

LEARNING-BASED MODELS

- **Learning from user interactions: Mitra (2015) continued**
 - ❏ Features to map syntactically and semantically similar query changes close together in the embedding space are also used in the learning model.
 - ❏ The training data is generated by sampling queries in the search logs and segmenting each query at every possible word boundary.
 - ❏ This approach provides a solution to recommend queries that have not been seen during the training period, an issue that could not be addressed by the previous approaches, whether heuristic or learning-based.

OTHER APPROACHES FOR RANKING CANDIDATE QUERIES

- **Sentence occurrence ranker:** Each query is scored based on the number of documents that match one or more of the query terms plus the number of documents that match all the query terms.
- **Time Ranker:** Each candidate completion is ranked based on the difference between the current time and the most recent past occurrence of it in the log, thereby promoting recent queries.
- **WordNet similarity ranker:** It uses WordNet in order to capture semantic similarities between the query being ranked and the previous queries entered by the same user. The similarity is calculated for every pair of terms present in the completion being ranked and the user context. Mean similarity over the user's previous queries is used for scoring the completion.

OTHER APPROACHES FOR RANKING CANDIDATE QUERIES (CONTD)

- **Kernel similarity ranker:** Collection enrichment using the corpus is applied to each candidate query and the user context. A candidate query is scored based upon the similarity between its expanded form and the expanded form of the user context.

Current search engines most probably use one or a combination of more than one of the discussed approaches in their autocomplete algorithms.



AUTOCOMPLETE FUNCTIONALITY CONCLUSION



ADVANTAGES

- Reduces search time by offering suggestions, which means the user will rarely have to enter their entire query
- Offers a more personalized experience, learned by users' previous queries, location, and preferred categories
- Shows users appropriate and useful suggestions based on popular searches
- Increases and improves the overall user experience by making content available in a more optimized fashion
- Increases the number of relevant queries, and offers optimized results
- Helps to avoid confusion caused by typos and errors, and offers faster results by filtering the data

DISADVANTAGES

- Can be a visually intense experience, possibly even an overwhelming one. Tech journalist John Pavlus described it as "like having a web search seizure. [The] screen explodes with noise as you type."
- Drives more traffic to the most statistically probable searches. The most-trafficked ways of searching for something will get more trafficked. Thus, the number of unique searches drop because people see something in the list that makes sense, even if it is not exactly how they would have worded it. This leads to influencing the behaviour of the people.

CONTROVERSIES

- In a case brought before the Tribunal de Grand Instance of Paris in 2010 and eventually decided by the Supreme Court, it was found that a suggested search indulges users' curiosity and orients them towards searches that may in turn influence the algorithm. Such a "snowball effect" goes against the "neutral algorithm" argument that was set forward by Google in its defence. Autocomplete could orient people towards searches they would not have otherwise performed.
- The French organization "SOS Racisme", along with five others, sued Google for hate speech in May 2012. According to the claim, suggestions had words like "Jew" or "Jewish" after the names of certain public figures, potentially exposing anti-Semitic hate speech.

CONTROVERSIES (CONTD)

- Guy Hingston, an Australian cancer surgeon, sued Google in Federal Court in December 2012. According to the complaint, when an individual computer user types 'Guy Hin ...', into the Google search engine as a search, the words 'Guy Hingston Bankrupt' appeared, even though the article(s) to which the user was directed had absolutely nothing to do with a bankruptcy associated with Dr. Hingston. Dr. Hingston lost a number of patients and financiers as a consequence of the reference on Google to a bankruptcy.
- Bettina Wulff, wife of the former German president, sued Google in 2012 to stop rumours about her private life. When her name was typed into the Google search engine, suggestions included "prostitute" and "red light district".
- In December 2016, Google announced that it had fixed a troubling quirk of its autocomplete feature: When users typed in the phrase, "are jews," Google automatically suggested the question, "are jews evil?"

WAYS IN WHICH A USER CAN CAUSE WORDS TO BE EXCLUDED (GOOGLE AUTOCOMPLETE)

1. If a search term prediction the user encounters is inappropriate, they can report it within the search box by clicking “Report inappropriate predictions”.
2. If a user has a legal objection to a search term prediction, they can fill out a report form and request its manual removal.

SUBVERTING AUTOCOMPLETE

Autocomplete has now become a part of reputation management as companies linked to negative search terms such as scam, complaints and fraud seek to alter the results. Google in particular have listed some of the aspects that affect how their algorithm works, but this is an area that is open to manipulation. In particular, Amazon's Mechanical Turk is a well-known venue where people can request that others do searches. When enough searches happen, suggestions start appearing. Brent Payne is probably one of the most notable examples of someone deliberately doing this "above the radar," so to speak. He ran a series of experiments where he hired people on Mechanical Turk to do searches, which (until Google removed them) impacted the suggestion feature.

SUBVERTING AUTOCOMPLETE (CONTD)

Do a search for 'brent payne manipulated this', click on any link, copy and paste first line of text from the page. EASY!


Requester: Brent D Payne

Qualifications Required: HIT approval rate (%) is not less than 90

When you do a search for 'brent payne manipulated this' on Google.com, what's the text of the first search result listed?

You must ACCEPT the HIT before you can submit the results.

Google

 Everything

 Images

 Videos

brent payne

brent payne

brent payne **drummer**

brent payne **manipulated this**

brent payne **avant**

brent payne **twitter**

CONCLUSION

Autocomplete makes it faster and easier for people to complete searches that they begin typing, thus improving the overall user experience. It has become an expected feature in search and has been implemented by almost all organizations, big or small. It takes into account the search history and interests of the user, thus producing query completions relevant and personalised to the user. Various approaches have been proposed for implementing the autocomplete feature. Despite the controversies which arose due to the predictions being based on real searches and word patterns found across the web, autocomplete has become an integral feature for search engines.

REFERENCES

- https://en.wikipedia.org/wiki/Search_engine
- <https://www.javatpoint.com/search-engines>
- <https://en.m.wikipedia.org/wiki/Autocomplete>
- <https://www.search.io/blog/predictive-search-and-autocomplete>
- <https://www.theatlantic.com/technology/archive/2013/08/how-googles-autocomplete-was-created-invented-born/278991/>
- <https://www.courthousenews.com/googles-search-box-defamed-him-doc-says/>
- <https://searchengineland.com/how-google-instant-autocomplete-suggestions-work-62592>
- <https://www.onely.com/blog/everything-about-google-autocomplete/>
- <https://www.theatlantic.com/technology/archive/2010/09/the-pros-and-cons-of-google-instant/62666/>
- <https://www.bbc.com/news/technology-19542938.amp>

REFERENCES

- Cai, F., & De Rijke, M. (2016). A survey of query auto completion in information retrieval. *Foundations and Trends® in Information Retrieval*, 10(4), 273-363.
- Di Santo, G., McCreddie, R., Macdonald, C., & Ounis, I. (2015, August). Comparing approaches for query autocompletion. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 775-778).
- Karapapa, S., & Borghi, M. (2015). Search engine liability for autocomplete suggestions: personality, privacy and the power of the algorithm. *International Journal of Law and Information Technology*, 23(3), 261-289.



Thank
You