

Micro-content Shortlisting using Interactive AI Method

Veeresh Devireddy

Former Cisco, Apple and Ebay Engineer, Bengaluru, India.

Abstract - A cross-domain real-time shortlisting method is provided using Artificial Intelligence (AI). The method includes instantaneous identification of relevant information on the user interface (UI) like the web browser and uses a system to inject actionable buttons in an overlay on the website for shortlisting. Further, the relevant information can be shortlisted dynamically while the user is browsing. Thereafter, the shortlisted information will be available in one place as a universal list and accessible across other websites with a consistent experience.

Keywords - shortlist, smartylist, microcontent, AI, UI, relevant, domain, smartlist.

1. INTRODUCTION

With the increasing penetration of the internet across the world, information available on the internet has exploded. User accumulates information from several sources on the internet for various purposes such as journalism, research, education, media coverage, social networking, teamwork collaboration, reviewing of shopping products, or to manage any data that they like or needed while browsing.

Currently, there is no convenient solution to manage the data efficiently and instantaneously in the above fields and contexts. Typically and most often, users would try to save, copy or manage the relevant information, either for local reference or share to collaborate with others. Some examples include saving search results, daily news, favoriting videos and photos, social feeds, products on e-commerce sites, and in general any text or links on the internet.

However, there is nothing to store all of the relevant data in one place as a common Shortlist that acts as a universal list. Also, there is no quick and intuitive way to access and review those sets of information among multiple platforms (websites), and yet not being able to control them in one place. In light of the above-mentioned drawbacks, there exists a need for a solution that allows a user to quickly save and manage multiple patterns of relevant information from various websites.

2. SUMMARY OF THE INVENTION

A primary object of the disclosure is to provide a method of shortlisting relevant information on the internet browser, using an Interactive AI technology. Another object of the present disclosure is to provide a method of

cross-domain information shortlisting on the internet browser, dynamically without affecting the browser performance, user experience and having control to manage them consistently on those cross-domain websites.

3. DETAILED DESCRIPTION

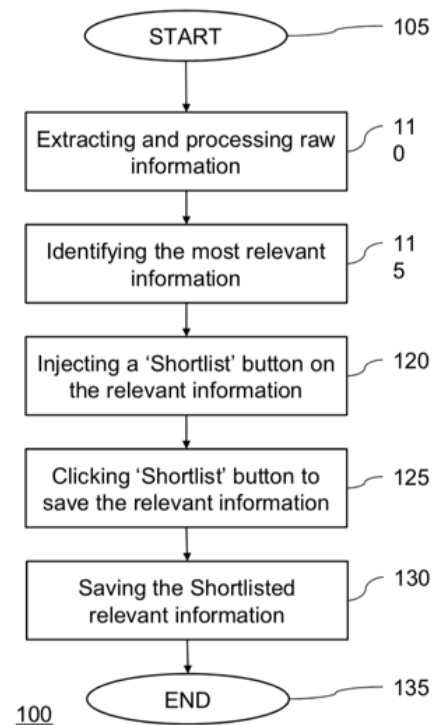


FIG. 1

Illustrates a method 100 of cross-domain shortlisting, in accordance with an embodiment of the present disclosure. The method 100 starts at step 105, with a user loading a website on the browser. At step 110, raw information from the website UI is extracted and processed. Further, at step 115, the most relevant information is identified, in parallel while the user is browsing.

In an embodiment of the present disclosure, the relevant information is based on the predefined domain (website name). The UI can be any website on the internet, and also applicable to any mobile, tablet or other electronic user interfaces. Examples of the relevant information can be daily news, favoriting videos and photos, social feeds, products on e-commerce sites, and in general any text or links on the internet.

The relevant information is unique in every domain. For example, the information on an e-commerce domain can be defined as product name, price, shipping time, ratings, and so forth. On Youtube, relevant information may be a video. Text and an image could be the relevant information on a news website. It should be understood that type of relevant information from each domain or a group of domains is predefined, and are identified by the Artificial Intelligence (AI) algorithms based on the abstract rules. Abstract Rules are simple parameters to an Artificial Intelligence system, which use a series of IF-THEN statements that guide a computer to identify, validate and extract information. In an embodiment of the present disclosure, the relevant information is identified by automatic detection based on one or more of HTML data from a server, structure of data, type of data, the position of data on the UI, loading type, user actions, and so forth.

Thereafter, at step 120, a 'Shortlist' button is injected on the relevant information. The user clicks on the 'Shortlist' button to save the relevant information at step 125. For example, a user is browsing an e-commerce website. While browsing, the user likes a product and wants to save it so that he can later compare it with other products on several e-commerce platforms. The user can click the 'Shortlist' button to save the product information.

The shortlisted relevant information can be a combination of a title, a short description, an image, a link, a price, rating, visits, etc. In another embodiment of the present disclosure, the shortlisted relevant information is tagged by the user. For example, a user may add a 'Home' tag to related products for the user's own home.

In an embodiment of the present disclosure, a 'Shortlist' button may dynamically appear on the relevant information on the UI. In another embodiment of the present disclosure, the user can select (by dragging the area) any information on the UI to shortlist, if the 'Shortlist' button is not visible. Here, the system may have ignored it if it doesn't contain all of the necessary meta-data, hence the user is performing this manual action. For this manual shortlisting process, the user needs to Right Click (Mouse) on the web UI and then select an option "Add to Shortlist".

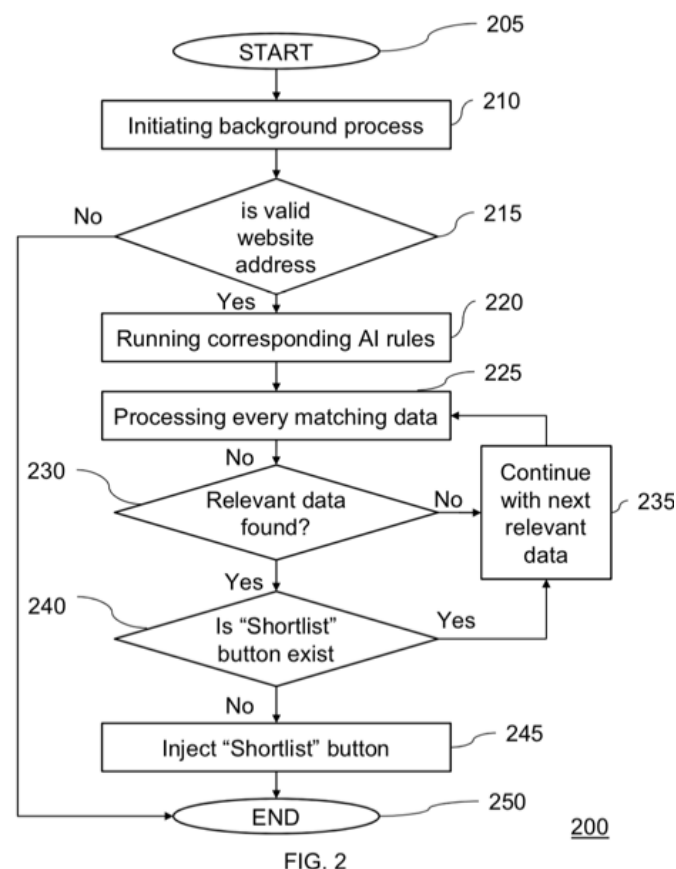
In an example, the system fails to find relevant information when there are no key data elements like Titles, Links, Image, Price tag, etc. For example, a news article without a title, a product without a price tag, or a broken link, etc.

Subsequently, at step 130, the shortlisted relevant information is saved in one place on UI as a universal list. This universal list is accessible across a plurality of domains. Referring to the previous example, the

shortlisted product information is saved in a universal shortlist. This universal shortlist is accessible across all platforms, devices, and domains.

For example, a user can shortlist a product on Amazon.com and it is also accessible on Flipkart.com while reviewing other products. In an embodiment of the present disclosure, this universal list is stored locally and then synced to cloud storage periodically. It should be noted that storing the universal list can be implemented in several ways without impacting the scope of the present disclosure.

In another embodiment of the present disclosure, the shortlist of items can be shared with other users for collaboration. For example, a user can share all shortlisted information having a 'personal' tag with his family members. In an embodiment of the present disclosure, the method 100 allows real-time tagging of text on browser UI by injecting an interface processor as a plug-in to save and manage the life cycle of the relevant information shortlisting as a universal list. The method 100 ends at step 135.



Illustrates a method 200 of data processing and shortlisting, in accordance with an embodiment of the present disclosure. The method 200 initiates at step 205. The background process (browser plug-in) is initiated at step 210. At step 215, it is checked whether the current website address is valid or not.

The validity of the website address is determined based on the syntax, format and if it supported by the system. If the website address is invalid, the method 200 ends at step 250. In case, if the website address is valid, corresponding Artificial Intelligence (AI) rules run at step 220.

For example, when the user browses an e-commerce website Amazon.com, then it's corresponding AI rules will get executed within the browser to find relevant products having several formats.

Subsequently, every matching data is processed at step 225. Here, the AI rules are pre-defined to find several closest matching data, but not all necessarily be relevant and valid. For instance, matching news data is duplicate, a product may not have a price or its picture. Hence, every matching data will be validated against all conditions and formats to be eligible to shortlist.

If no relevant data is found at step 230, the method 200 continues with the next relevant data at step 235. Subsequently, every matching data is processed at step 225.

In case relevant data is found at step 230, it is checked whether a 'Shortlist' button exists at step 240. If the 'Shortlist' button is found, the method 200 goes back to step 235, because the matched relevant data already identified and has its 'Shortlist' button. If the 'Shortlist' button is not found at step 240, then it continues with step 245 to inject the new 'Shortlist' button on the browser UI where the relevant data is found. Thereafter, the method 200 ends at step 250.

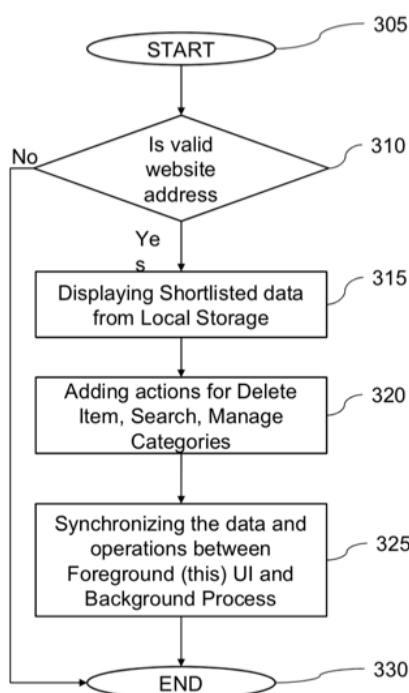


FIG. 3

300

Illustrates a method 300 of displaying shortlisted data as an overlay on the website within the browser UI, in accordance with an embodiment of the present disclosure. The method 300 initiates at step 305.

At step 310, it is checked if a valid website address is used. If a valid website address is not used, the method 300 ends at step 330. If the valid website address is used, shortlisted data from local storage is displayed at step 315. Further, actions for 'Delete Item', 'Search', 'Manage Categories' and so forth are added at step 320. These are actions or functions offered to a user to manage Shortlisted data on the same page. At step 325, the data and operations are synchronized between foreground UI and background process.

For example, any user actions on UI like clicking on the 'Shortlist' button on the relevant content, changing the category of the shortlisted item, adding a new category, deleting the shortlisted item, etc. will push the updates into the background process.

Similarly, any changes in the configuration, importing of shortlisted items will pull the changes from the background process to UI on the foreground. Further, the method 300 ends at step 330.

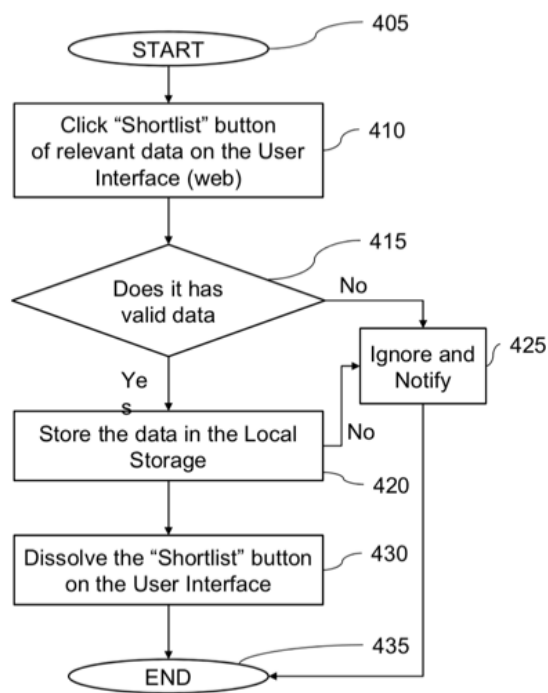


FIG. 4

400

Illustrates a method 400 of adding an item into Shortlist, in accordance with an embodiment of the present disclosure.

The method 400 initiates at step 405. A user clicks on the "Shortlist" button of the relevant data on the browser UI. For example, the "Shortlist" buttons are placed on the elements or container of the relevant data on the website. For instance, this button can be placed within any HTML tags like Division (<div />), Paragraph (<p />), Span () etc, and the "Shortlist" buttons will have an overlay effect so the user can click them independently. It is checked at step 415, whether the "Shortlist" button contains the necessary valid data. If it has invalid data or failed to get stored, then it will be ignored and notified to the user at step 425. Subsequently, the method 400 ends at step 435.

If valid data is found at step 415, the data is stored in the local storage at step 420. For example, when we shortlist a product on an e-commerce site, then its Title, Link, Picture, Price will be stored in the local storage with its unique ID. If the storage fails at step 420, it is ignored and notified to the user at step 425. If successfully stored, then the "Shortlist" button of that particular relevant data item gets dissolved on the UI at step 430, mainly to indicate to the user that it has already got shortlisted. The method 400 subsequently ends at step 435.

4. FINAL CLAIMS

1. A cross-domain shortlisting method comprising:

- A. Live extracting and processing of information on the web UI to find the potential relevant data of the user using the Artificial Intelligence (AI) approach, wherein the UI is an internet browser, either on a desktop computer, mobile or tablets, and its functionalities are limited to a set of predefined website addresses (domain);
- B. Identifying the relevant information based on the factors like relevance, format, pattern, type, and nature of data;
- C. Injecting the "Shortlist" buttons within the website when the relevant data is identified while the user is browsing;
- D. Instant shortlisting of relevant data by clicking on the "Shortlist" buttons on the UI of the website; and
- E. Saving the shortlisted information as a universal list and making them accessible across a plurality of domains.

2. The cross-domain shortlisting method as claimed in claim 1, wherein the relevant information is identified and extracted based on the predefined factors and website domains.

3. The cross-domain shortlisting method as claimed in claim 1, wherein the shortlisted relevant information has one or more of a title, a short description, an image, a link, a price, rating, and visits.

4. The cross-domain shortlisting method as claimed in claim 1 further comprising tagging the shortlisted relevant information by one or more tags, wherein the one or more tags are defined by the user.

5. The cross-domain shortlisting method as claimed in claim 1, wherein the one or more universal lists are stored locally and periodically synced to cloud storage.

6. The cross-domain shortlisting method as claimed in claim 1, wherein identifying the relevant information comprises automatic detection based on one or more of HTML data from the website server, the structure of data, type of data, the position of data on the UI, loading type, and user actions.

7. The cross-domain shortlisting method as claimed in claim 1, wherein shortlisting the relevant information comprises manually selecting or dragging the text on the web UI and then manually shortlisting it.

8. The cross-domain shortlisting method as claimed in claim 1, wherein the predefined domain represents the IP address of the remove website server.

9. The cross-domain shortlisting method as claimed in claim 1 further comprising real-time tagging, wherein the real-time tagging comprises:

- A. Injecting an interface process, wherein the interface process is a program that interacts between a browser's background plug-in and the foreground website running on the same browser UI, in order to process website data;
- B. Detecting relevant data; and
- C. Shortlisting the detected relevant data.

5. CONCLUSION

This method is going to revolutionize the way we use internet data, by improving the most fundamental experience of managing live-data on web user interfaces like websites. It's Interactive AI method will help auto-detect the most relevant data to shortlist and manage them intuitively, not only on the web but pretty much on any interfaces, like Mobile, Tablets, TV, etc. It can be integrated with multiple websites and enriches the UI to manage data efficiently with a consistent experience. This solution will offer a free-style control to manage the data, and making them accessible even on other websites, as a brand new Universal List, helping global users' day-to-day use cases.

REFERENCES

- [1] B. Adelberg. NoDoSE – a tool for semi-automatically extracting structured and semistructured data from text documents. In SIGMOD'98.
- [2] E. M. Gold. The complexity of automaton identification from given data. *Information and Control*, 37(3), 1978.
- [3] Valter Crescenzi. *Towards Automatic Data Extraction from Large Web Sites*. 2001
- [4] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118, 2000.
- [5] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3), 1999.
- [6] Dayne Freitag, *Machine Learning for Information Extraction in Informal Domains*, *Machine Learning*, v.39 n.2-3, p.169-202, May 2000
- [7] MUSLEA, I. RISE: Repository of online information sources used in information extraction tasks. <http://www.isi.edu/muslea/RISE/>
- [8] Serge Abiteboul, *Querying Semi-Structured Data*, *Proceedings of the 6th International Conference on Database Theory*, p.1-18, Jan 08-10, 1997
- [9] Alberto H. F. Laender. A brief survey of web data extraction tools. *ACM SIGMOD*, June 2002
- [10] D.W.Embley. *Conceptual-model-based data extraction from multiple-record Web pages*. Elsevier, Vol 31, Nov 1999

BIOGRAPHY

I was born in Karnataka, India, and got a Bachelor's Degree and Diploma in Computer Science from Visvesvaraya Technological University. Then moved to California, USA and gained a significant experience at Cisco, Apple and Ebay. Now, Founder at Smartylist.com.