APPLICATION SIMILARITY

Lei Yang
Zhikun Wang
Zhen Hao Zhou
Jiahui Liu
Zhen Xu

See next page for additional authors

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
Yang, Lei; Wang, Zhikun; Zhou, Zhen Hao; Liu, Jiahui; Xu, Zhen; Li, Zhen; Lee, Hsu-Chieh; and O’Malley, Sean, "APPLICATION SIMILARITY", Technical Disclosure Commons, (August 03, 2018)
https://www.tdcommons.org/dpubs_series/1588

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
APPLICATION SIMILARITY

ABSTRACT

An application (app) store system may include functionality to determine whether an application is similar to other applications in the store. The application store system may determine similarity based on a number of aspects, such as title, description, icon, and application programming code.

DESCRIPTION

Application stores provide a way for application developers to disseminate their applications to a user base. An application store may provide an indication of a user rating, a number of downloads, user comments, images, videos, description, related applications, etc. In some instances, the application store may generate a recommendation for one or more applications based on a user’s purchase history, the user’s browsing history, the purchases of the user’s friends, downloads, ratings of applications, etc. However, existing application store systems may not have the ability to determine whether applications are similar to one another. As such, existing application stores may not be able to take action (e.g., recommending an application or other actions) based on application similarity. As such, it would be desirable for an application store system to be able to determine application similarity.

The example application store system shown in Figure 1 has the ability to determine whether an application is similar to other applications. For example, the application store system may determine whether Application_A is similar to any of the applications stored in the Asset Repository (i.e., similar to other applications known to or available in the application store).
The application store system of Figure 1 includes a similarity classifier, an icon similarity calculator, a title similarity calculator, a developer name similarity calculator, and an asset repository. It is understood that the application store system may further include devices/modules/elements that enable users to view/purchase/download/search applications. The system may represent any type of server, cloud computing environment, or other computing system that is configured to determine application similarity.

Similarity classifier may receive an application, such as Application_A, and determine whether the received application is similar to any of the applications stored in the asset repository. The similarity classifier may determine application similarity based on various features of the received application. In the example of Figure 1, the similarity classifier may determine application similarity based on an icon, title, and developer name of the received application, although other features may be used in addition to or in place of icon, title, and developer name.

The application store system may include one or more calculators that may calculate a similarity score for each of the application features. As shown in Figure 1, the application store
system may include an icon similarity calculator, a title similarity calculator, and a developer name similarity calculator.

The similarity classifier may provide the relevant features of the application under consideration to the various calculators. For instance, the similarity classifier may provide an icon of Application_A to the icon similarity calculator, a title of Application_A to title similarity calculator, and a developer name of Application_A to developer name similarity calculator.

Each of the calculators may determine, based on the received feature(s) of the application under consideration, similarity scores that represent a degree of similarity between the received feature(s) and corresponding feature(s) of applications in the asset repository. The calculators may return (e.g., to the similarity classifier) features of the top N (e.g., 1, 2, 3, 4, 5, 10, 20, 50, etc.) most similar application in the asset repository along with the similarity scores. For instance, the icon similarity calculator may compare the icon of Application_A with icons of respective applications in the asset repository to generate respective scores that indicate how similar the icon of Application_A is to the icons of the respective applications in the asset repository, and return the top N most similar icons and corresponding scores to the similarity classifier. The title similarity calculator and the developer name calculator may perform similar operations respectively with titles and developer names.

As discussed above, the icon similarity calculator may compare the icon of the application under consideration with icons of applications in the asset repository. In some examples, the icon similarity calculator may use a machine learning model (e.g., a convolutional neural network) to perform the comparisons and/or generate the scores. The machine learning
model may be trained using image triplets (e.g., \(<\text{anchor image}, \text{positive image}, \text{negative image}>\) with the following loss function\(^1\):

\[
L = \sum_i \left( 0.5 \cdot \| f_a - f_p \|^2 - 0.5 \cdot \| f_a - f_n \|^2 + g\text{ap}_\text{param} \right) + \alpha \cdot 0.5 \cdot \| f_a - f_p \|^2
\]

where \( f(x) \) is the embedding for exemplar \( x \), \( a \) is anchor image, \( p \) is positive image, \( n \) is negative image, and \( \alpha \) is the weight for the additional positive contraction term.

As discussed above, the title similarity calculator may compare the title of the application under consideration with titles of applications in the asset repository. In some examples, the title similarity calculator may perform the comparisons and/or generate the scores based on a normalized edit distance (e.g., with pre-processing) between the title of the application under consideration and titles of application in the asset repository, text embedding distance between the title of the application under consideration and titles of application in the asset repository, whether the title of the application under consideration and titles of application in the asset repository have common knowledge graph term extracted, and/or any other suitable technique.

The similarity classifier may receive the results of the calculations from the feature calculators. For instance, the similarity classifier may receive the top \( N \) similar icons and corresponding scores (e.g., the respective scores that indicate how similar each of the respective \( N \) similar icons are to the icon of the application under consideration) from icon similarity calculator, the top \( N \) similar titles and corresponding scores (e.g., the respective scores that indicate how similar each of the respective \( N \) similar titles are to the title of the application under consideration) from title similarity calculator, and the top \( N \) similar developer names and corresponding scores (e.g., the respective scores that indicate how similar each of the respective

\(^1\) See e.g., https://en.wikipedia.org/wiki/Loss_functions_for_classification
N similar developer names are to the developer name of the application under consideration) from developer score similarity calculator.

The similarity classifier may determine, based on the received results of the calculations, whether or not the application under consideration is similar to at least one of the applications in the asset repository. In some examples, the determination may be binary in that the similarity classifier may generate a binary value that indicates whether the application under consideration is similar to at least one of the applications in the asset repository. In some examples, in addition to or in place of the binary determination, the similarity classifier may also output an indication of which application(s) in the repository the application under consideration was found to be similar. Where the similarity classifier output the indication of which application(s) in the repository the application under consideration was found to be similar, the similarity classifier may also output the similarity scores (e.g., the scores generated by the feature calculators).

Examples are described wherein a computing device and/or computing system may analyze information (e.g., e-mail, other communications, and the like) associated with a user of the computing device (e.g., application developer) only if the computing device and/or the computing system receives explicit permission from the user of the computing device to analyze the information. For example, in situations in which the computing device and/or computing system may collect or may make use of communication information associated with the user and the computing device, the user may be provided with an opportunity to provide input to control whether programs or features of the computing device and/or computing system can collect and make use of user information (e.g., information about a user’s e-mail, a user’s social network, social actions or activities, profession, a user’s preferences, or a user’s past and current location), or to dictate whether and/or how the computing device and/or computing system may receive
content that may be relevant to the user. In addition, certain data may be treated in one or more ways before it is stored or used by the computing device and/or computing system, so that personally-identifiable information is removed. For example, a user’s identity may be treated so that no personally identifiable information can be determined about the user, or a user’s geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over how information is collected about the user and used by the computing device and/or computing system.